



Uniwersalna platforma chmurowa w
zastosowaniach Internet Of Things

Warsztaty techniczne
Skrypt do ćwiczeń

2019-04-12, wersja pierwsza

Autor: Maciej Szulc, IBM, maciej.szulc@pl.ibm.com

<u>ACHITEKTURA TWORZONEGO ROZWIĄZANIA</u>	3
<u>PRZYGOTOWANIE ŚRODOWISKA CLOUD</u>	5
1. URUCHOMIENIE USŁUGI WATSON IoT PLATFORM	5
2. URUCHOMIENIE INSTANCJI NODE-RED	7
3. PIERWSZE KROKI W NODE-RED	9
<u>WATSON IOT</u>	14
4. KONFIGURACJA PLATFORMY	14
5. POWIĄZANIE USŁUG W IBM CLOUD	19
6. WYŁĄCZENIE WYMOGÓW WZGLĘDEM SZYFROWANIA POŁĄCZEŃ	23
<u>SYMULATOR URZĄDZENIA</u>	25
7. UTWORZENIE SYMULATORA IoT	25
8. WERYFIKACJA KOMUNIKACJI	34
<u>WIZUALIZACJA DANYCH POMIAROWYCH</u>	36
10. UTWORZENIE PROSTEJ WIZUALIZACJI DANYCH W PLATFORMIE IoT	36
11. DODANIE SKŁADOWANIA DANYCH HISTORYCZNYCH	41
12. SPRAWDZENIE ZAWARTOŚCI BAZY DANYCH PRZECZEWUJĄcej DANE HISTORYCZNE.	43
<u>WIZUALIZACJA DANYCH POMIAROWYCH W NODE-RED DASHBOARD</u>	46
13. INSTALACJA PAKIETU „NODE-RED-DASHBOARD”	46
14. UTWORZENIE PODSTAWOWEJ WIZUALIZACJI	48
15. SYMULOWANIE OTWARCIA I ZAMKNIĘCIA DRZWI	63
<u>ZADANIA DODATKOWE – DLA CHĘTNYCH</u>	69
16. NUMER KARTY	69

ACHITEKTURA TWORZONEGO ROZWIAZANIA

Niniejsze ćwiczenie będzie dotyczyło rejestracji i prezentacji informacji z sensora IoT, komunikującego się z platformą IBM Cloud poprzez niezwykle popularny w zastosowaniach IoT protokół MQTT.

Nasz sensor będzie nadzorował pomieszczenie serwerowni cyklicznie podając warunki środowiskowe oraz przekazując informacje związane z bezpieczeństwem.

Ze względu na możliwości techniczne i czasochłonność zadania nie będziemy używali rzeczywistego urządzenia IoT, lecz będziemy go symulować poprzez odpowiednie programy działające w środowisku chmurowym. Mimo tego od strony technicznej nasze symulowane urządzenie będzie w pełni odzwierciedlało wszystkie czynności wymagane do podłączenia prawdziwego, sprzętowego urządzenia.

W trakcie wykonywania poszczególnych kroków wykorzystamy kilka usług i aplikacji dostępnych w chmurze:

- **Internet Of Things Platform** – wykorzystywana w naszym ćwiczeniu do odbierania i wysyłania danych pomiarowych, a także do prostej ich wizualizacji,
- **Node-Red** – którą użyjemy na dwa sposoby. Pierwsza funkcja to wspomniany już symulator urządzenia. Druga – to wizualizacja zebranych danych,
- **Cloudant NoSQL DB** – baza danych niezbędna dla prawidłowej pracy Node-Red i przechowująca zebrane dane historyczne

Symulator, który przygotujemy będzie generował dane pomiarowych - komunikat o nazwie „status” i formacie:

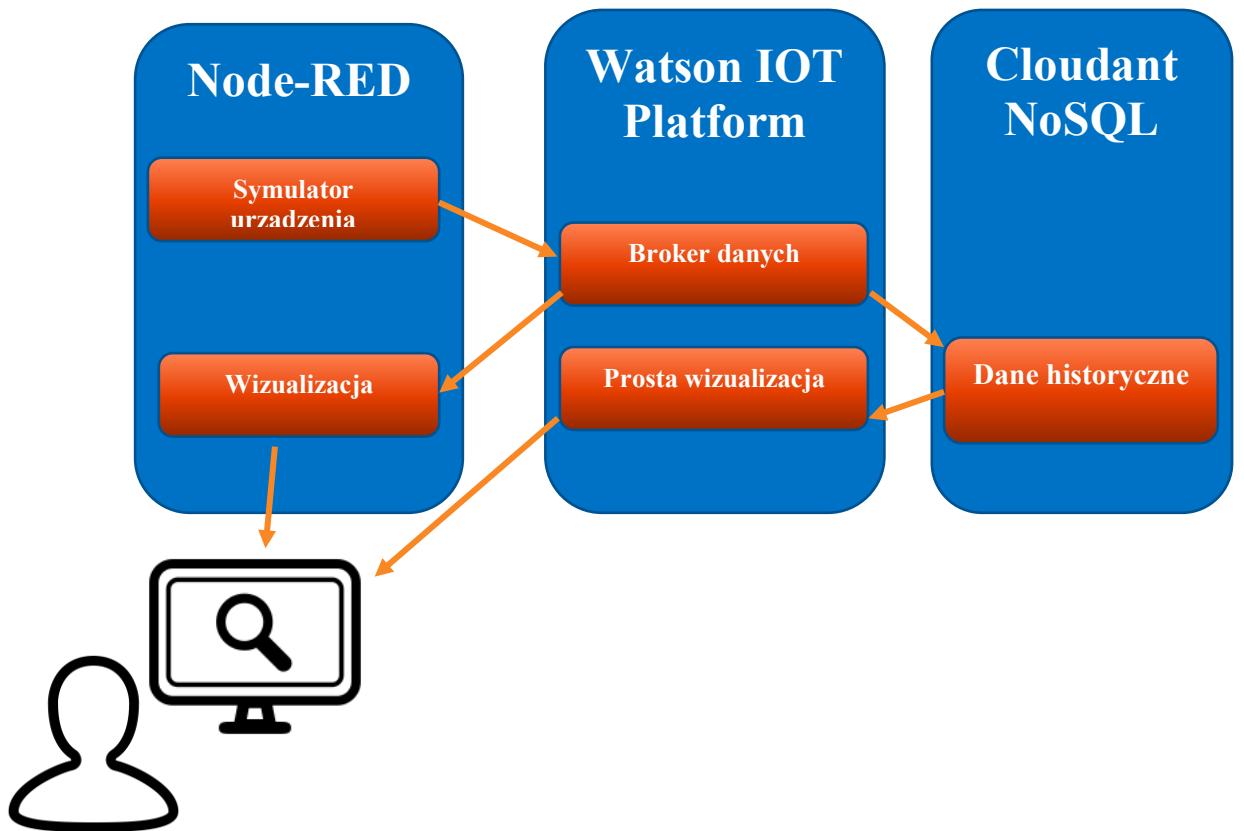
```
"d": {  
    "timestamp": 1555321311294,  
    "temp": 25.62,  
    "humi": 85.13,  
    "doorA": 2,  
    "doorB": 2,  
    "doorC": 2  
}
```

Gdzie poszczególne pola mogą przyjmować wartości:

Obiekt	Wartość	Znaczenie
doorA	0	Czujnik wyłączony
	1	Otwarte
	2	Zamknięte
doorB	0	Czujnik wyłączony
	1	Otwarte
	2	Zamknięte
doorC	0	Czujnik wyłączony
	1	Otwarte
	2	Zamknięte
Timestamp	Różna	Timestamp komunikatu w formacie Date().getTime()
Temp	Różna	Temperatura
Humi	Różna	Wilgotność

Dane te będą następnie przetwarzane i składowane przez komponenty tworzące budowane rozwiązanie.

Schemat działania naszego systemu będzie następujący:



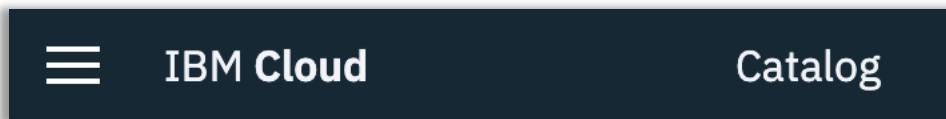
PRZYGOTOWANIE ŚRODOWISKA CLOUD

Pierwszy krok to utworzenie instancji usługi IBM Cloud Watson IOT Platform i odpowiednie jej skonfigurowanie.

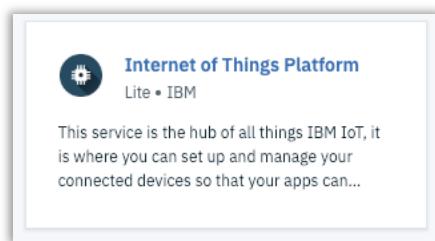
Musimy również zdefiniować w platformie IoT nowy typ urządzenia i uzyskać dane niezbędne do połączenia symulowanego czujnika.

1. URUCHOMIENIE USŁUGI WATSON IOT PLATFORM

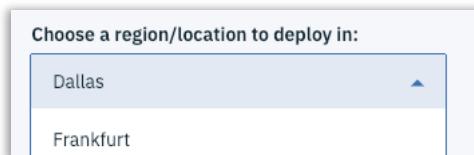
1. Zaloguj się do konsoli IBM Cloud (bluemix.net).
2. Kliknij na przycisk Catalog – lewy górny róg



3. Wyszukaj „Internet of Things Platform” Lite IBM, kliknij na usługę



4. Wybierz region „Frankfurt” (o ile Twój typ konta pozwala na wybór lokalizacji)



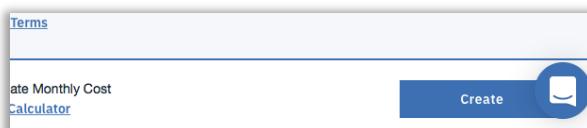
5. Znajdź sekcję “Pricing Plans” i sprawdź czy wybrany jest plan “Lite”

Pricing Plans		
Monthly prices shown are for country or region: Poland		
PLAN	FEATURES	PRICING
✓ Lite	Includes up to 500 registered devices, and a maximum of 200 MB of each data metric Maximum of 500 registered devices Maximum of 500 application bindings Maximum of 200 MB of each of data exchanged, data analyzed and edge data analyzed	Free

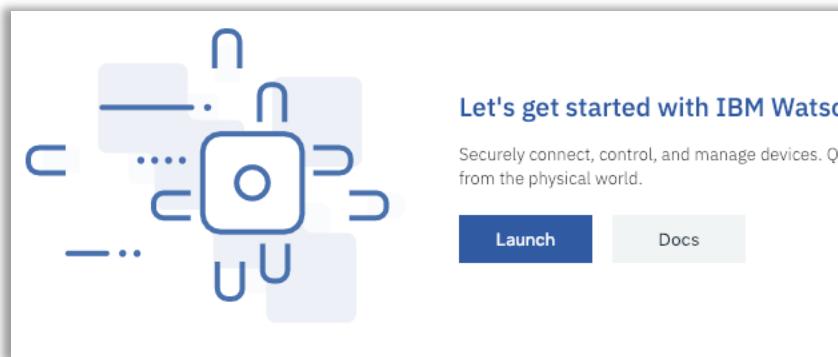
The Lite service plan for Internet of Things Platform includes up to 500 registered devices, and a maximum of 200 MB each of data exchanged, data analyzed, and edge data analyzed per month.

Lite plan services are deleted after 30 days of inactivity.

- Kliknij w przycisk „Create” na dole strony. Poczekaj chwilę aż usługa się utworzy.



- Uruchom dashboard usługi przyciskiem „Launch”.



- Ekran Watson IOT platform potwierdza, iż usługa została uruchomiona.

A screenshot of the 'IBM Watson IoT Platform' interface. The top navigation bar has tabs 'Browse', 'Action', and 'Device Types'. On the left, there is a vertical sidebar with icons for device management. The main content area is titled 'Browse Devices' and contains a summary table. The table has columns for 'All Devices' and 'Diagnose'. A note below the table says: 'This table shows a summary of all devices that have been added. It can be filtered using different criteria. To get started, you can add devices by using the Add Device button.' The 'Browse' tab is currently selected.

2. URUCHOMIENIE INSTANCJI NODE-RED

W naszej pracowni nie posiadamy urządzeń IoT które mogłyby wysyłać dane pochodzące z prawdziwych sensorów.

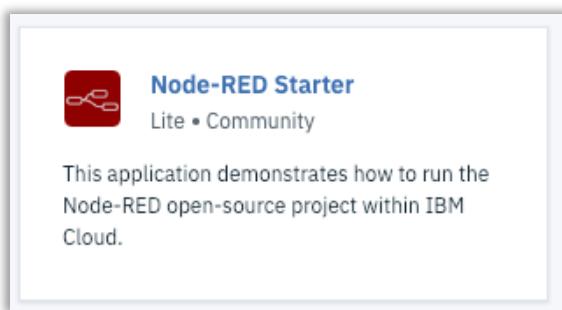
W dalszej części ćwiczenia skorzystamy zatem z możliwości platformy Node-Red którą użyjemy zarówno jako symulator urządzenia, jak i do utworzenia ekranów użytkownika. Node-RED to napisane w node.js graficzne i intuicyjne środowisko programistyczne, w którym za pomocą predefiniowanych bloków tworzymy algorytm programu. Projekt rozwija się bardzo dynamicznie, ma dużą społeczność - a co za tym idzie wiele różnych rozszerzeń, oprócz gotowych bloków mamy możliwość pisania własnego kodu opartego o Java Script.

Node-Red zostanie zainstalowany na platformie chmurowej IBM Cloud.

9. Pozostaw otwartą kartę z usługą IoT Platform a w nowej zakładce przeglądarki wejdź na stronę bluemix.net i zaloguj się do swojego środowiska (ew. Użyj karty z już otwartym interfejsem IBM Cloud).
10. Kliknij na przycisk Catalog – lewy górny róg



11. Wyszukaj „Node-RED Starter” Lite Community, kliknij na usługę



12. Wypełnij pole „App Name” oraz „Host Name”.

„App name” ustaw zgodnie ze Swoim wyborem.

„Host Name” wypełnij w formie „uw-imię-dzień_urodzin”, np.: uw-maciej-18. (identyfikator musi być unikalny w skali całej platformy – będzie używany jako nazwa domenowa tworzonej usługi)

Pozostałe pola pozostaw bez zmian i kliknij „Create”

App name:
Nadzór pomieszczeń

Host name:
uw-maciej-18

13. Poczekaj aż usługa wystartuje (kilka minut)

The first screenshot shows the application 'Nadzór pomieszczeń' in the 'Starting' state. The second screenshot shows the same application now in the 'Running' state.

14. Kliknij w link „Visit App URL”

15. Na ekranie „Welcome to your new Node-RED instance on IBM Bluemix” kliknij “Next”

16. Utwórz nowego użytkownika, zapamiętaj lub zapisz nazwę konta i hasło. Dane te będą potrzebne do zalogowania się do Twojej instancji node-red.

Secure your Node-RED editor

Secure your editor so only authorised users can access it

Username: mszulc

Password: strong

Allow anyone to view the editor, but not make any changes

17. Na ekranie “Browse available IBM Bluemix nodes” wybierz “Next”

18. Zakończ konfigurowanie usługi poprzez “Finish”. Poczekaj na zakończenie konfiguracji.

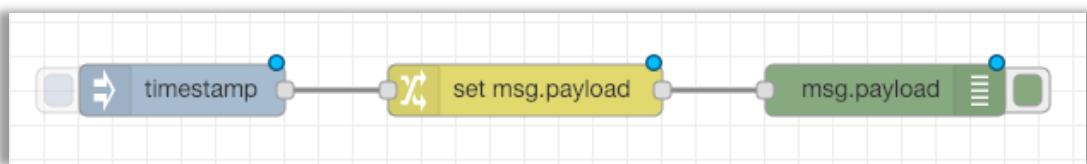
19. Zaloguj się do Twojej instancji Node-RED klikając „Go to your Node-RED flow editor”.

3. PIERWSZE KROKI W NODE-RED

20. Przeciągając ikonki z menu po lewej stronie utwórz program jak poniżej używając funkcji:

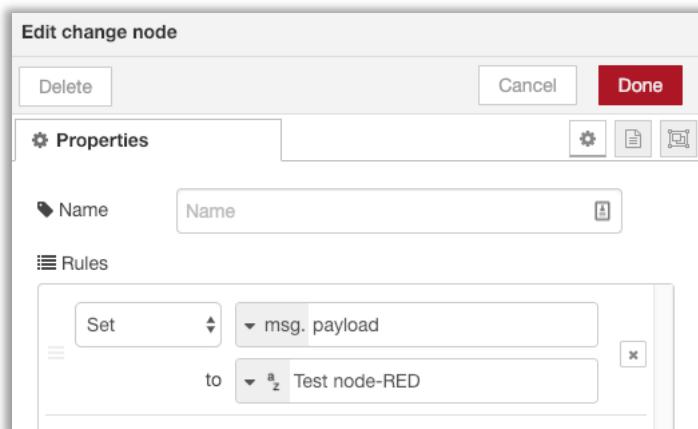
- Inject
- Change
- debug

oraz łączenia węzłów ze sobą (kwadraciki na ich końcach)

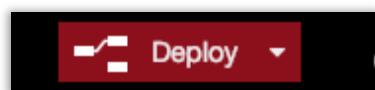


21. Większość węzłów posiada możliwość konfiguracji, odpowiednio do pełnionej w środowisku funkcji.

Kliknij dwukrotnie na węźle Change (set msg.payload) i ustaw jak poniżej, następnie zaakceptuj zmiany poprzez „Done”:



22. Uruchom program poprzez „Deploy”



23. Pod przyciskiem Deploy znajdziesz ikonkę z robaczkiem, służącą do debugowania programu. Kliknij ją.



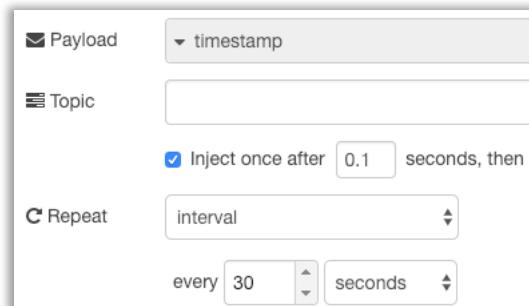
24. Zainicjuj przepływ klikając na „poduszce” węzła Inject (timestamp)



25. Zweryfikuj efekt w okienku „debug”

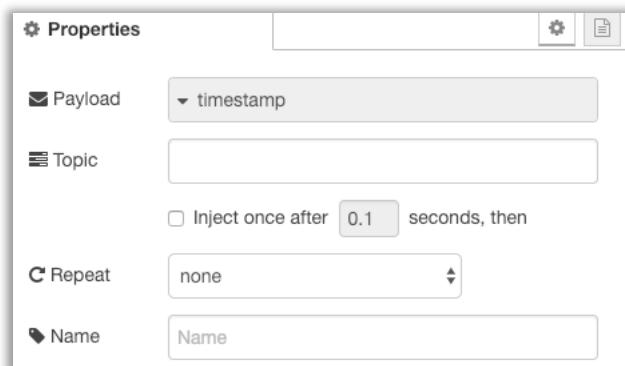
```
12.04.2019, 10:03:40  node: d29eda25.d37db8
msg.payload : string[13]
"Test node-RED"
```

26. Ustaw węzeł Inject (timestamp) tak, aby uaktywniał się co 30 sekund.



27. Uruchom kod i sprawdź poprawność działania programu.

28. Wyłącz automatyczne generowanie komunikatu w węźle Inject



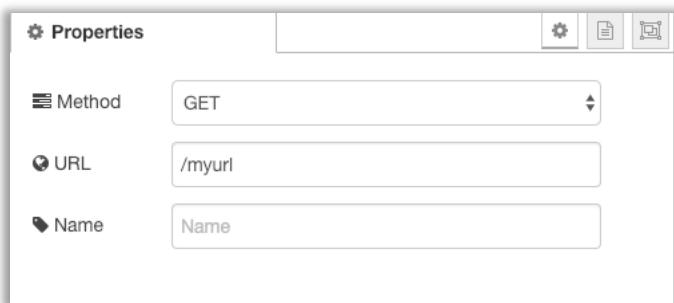
29. Teraz spróbujemy udostępnić nasz program dla przeglądarek www. W tym celu do programu dodaj węzły:

- http z grupy input

- http response z grupy output
i połącz je jak poniżej.

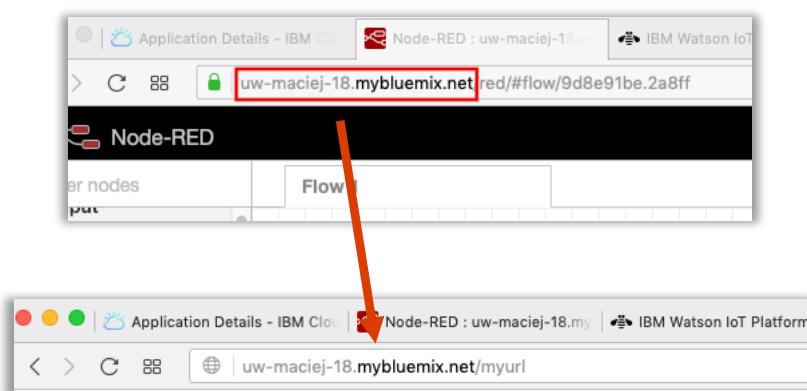


30. Skonfiguruj węzeł http jak poniżej:



31. Uruchom program przyciskiem **Deploy**

32. W nowej zakładce przeglądarki wejdź na url Twojej instancji Node-RED z dodaną końcówką „/myurl”:

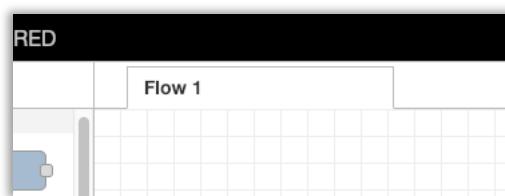


33. Sprawdź, czy program działa prawidłowo

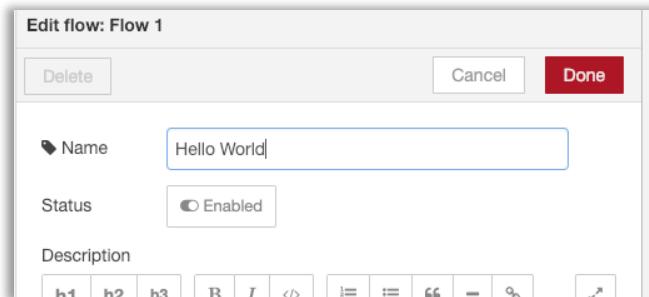


34. Powróć do zakładki z Twoim edytorem Node-RED

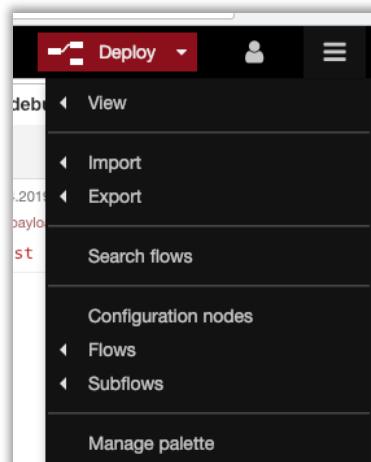
35. Zmieńmy nazwę programu: kliknij dwukrotnie na zakładce „Flow 1”



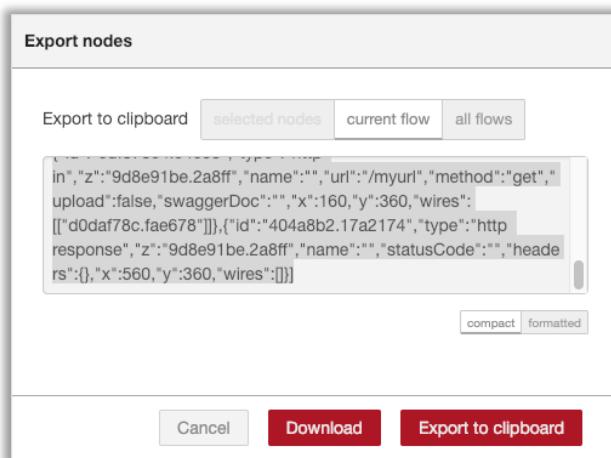
36. Nadaj nazwę programowi, kliknij „Done”



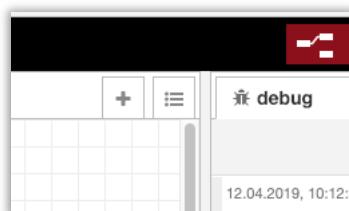
37. Tworzone programy możemy importować i eksportować. Aby to uczynić wybierz ikonkę „hamburgera” z prawej strony okna edytora i wybierz Export-> Clipboard



38. Zamknij okienko eksportu przyciskiem „Cancel”



39. Nowy program tworzy się w nowej zakładce Node-RED. Aby ją utworzyć kliknij „+” z prawej strony ekranu edytora:



40. Zmień nazwę nowego programu na „Symulator”.

41. Na tym kończymy wstępne zapoznanie się z funkcjonowaniem środowiska Node-RED.

WATSON IOT

4. KONFIGURACJA PLATFORMY

Platforma Watson IOT umożliwia połączenie różnego rodzaju obiektów – urządzeń, bram, aplikacji. My wykorzystamy tryb połączenia zwykłego urządzenia – gdzie wymagana jest wcześniejsza jego rejestracja w systemie. Krok ten można wykonać również poprzez API lub wręcz automatycznie, dzięki innym trybom połączania. My jednak wszystkie czynności zrobimy własnoręcznie.

Dodawanie nowego urządzenia będzie składało się z dwóch kroków:

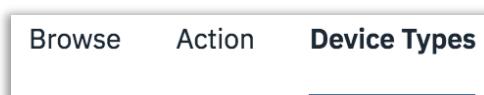
- Zdefiniowanie nowego typu urządzenia
- Zdefiniowanie samego urządzenia

42. Przejdz do zakładki z IBM Watson IoT Platform. Jeśli jej nie widzisz, to możesz do niej przejść z głównego interfejsu IBM Cloud („hamburger” po lewej stronie -> Dashboard -> CloudFoundry Services -> Internet of Things Platform -> Launch)



43. Korzystając z lewego paska wybierz menu „Devices”

44. Przejdz do zakładki „Device Types”



45. Zainicjuj dodawanie nowego urządzenia przyciskiem „Add Device” (prawy górny róg ekranu)



46. Jako rodzaj wybierz "Device" („Gateway” jest przeznaczone dla hubów IoT)

47. Nazwij i opisz swoje urządzenie, kliknij „Next”

Type	<input checked="" type="radio"/> Device	Or	<input type="radio"/> Gateway
Name	devSensor		
	The device type name is used to identify the device type uniquely and uses a restricted set of characters to make it suitable for API use.		
Description	Sensor środowiskowy		

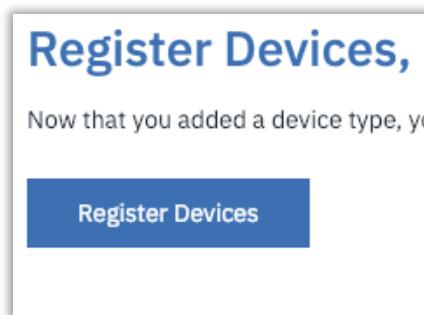
48. Wypełnij „Device Information”, kliknij „Done”

You can enter more information about the device type for identification purposes.

Serial Number	Enter Serial Number	⊕	Manufacturer	UW
Model	Enter Model		Device Class	Sensor środowiskowy
Description	Node-red symulator		Firmware Version	1.0
Hardware Version	Enter Hardware Version		Descriptive Location	Enter Descriptive Location
+ Add Metadata				

49. Kolejny krok do dodanie naszego sensora.

Pozostając w utworzonym typie urządzenia dodamy konfigurację – kliknij „Register Device”



50. Pozostaw uprzednio utworzony typ urządzenia i dodaj nowy deviceID – jego identyfikator.

Możesz go ustawić dowolnie, zgodnie z fantazją lub pozostawić jak poniżej.

Następnie kliknij „Next”.

Add Device **Identity** Device Information Security Summary

Select a device type for the device that you are adding and give the device a unique identifier.

Device Type	devSensor
Device ID	LabRoom-s1

51. Nie będziemy używać dodatkowych informacji o czujniku – na ekranie „Device Information” kliknij „Next”

52. W kolejnym kroku (Device Security) wybierz opcję tokena wpisywanego ręcznie (Self-provided authentication token) wpisując odpowiednie hasło w pole „Authentication Token”.
Jako token podaj dowolny ciąg znaków.
Kliknij „Next”

Add Device Identity Device Information **Security** Summary

There are two options for selecting a device authentication token.

Auto-generated authentication token (default) Allow the service to generate an authentication token for you. Tokens are 18 characters and contain a mix of alphanumeric characters and symbols. The token is returned to you at the end of the device registration process.	Self-provided authentication token Provide your own authentication token for this device. The token must be between 8 and 36 characters and contain a mix lowercase and uppercase letters, numbers, and symbols, which can include hyphens, underscores, and periods. Do not use repeated characters, dictionary words, user names, or other predefined sequences.
---	--

Authentication Token gory-sa-piekne-i-wysokie ⓘ

Make a note of the generated token. Lost authentication tokens cannot be recovered. Tokens are encrypted before being stored.

53. Zatwierdź dodawanie nowego urządzenia przyciskiem „Done”

Koniecznie zapisz uzyskane informacje.

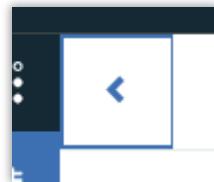


Dane z tabelki będą nam potrzebne w kolejnych krokach aby połączyć symulowane urządzenie!



Organization ID	kmymla
Device Type	devSensor
Device ID	LabRoom-s1
Authentication Method	use-token-auth
Authentication Token	gory-sa-piekne-i-wysokie

54. Zamknij okienko dodawania urządzenia – kliknij „Back” w lewym górnym rogu ekranu.



55. Kliknij na dodanym urządzeniu.

Browse Devices

All Devices Diagnose

This table shows a summary of all devices that have been added. It can be filtered, organized, and searched on using different criteria. To get started, you can add devices by using the Add Device button, or by using API.

<input type="checkbox"/>	Device ID	Device Type	Class ID	Date Added
<input type="checkbox"/>	LabRoom-s1	devSensor	Device	12 kwietnia 2019 12:35

56. Sprawdź stan dostępności – „Connection Status”

Usługa IoT Platform automatycznie zmieni stan urządzenia, gdy tylko się ono podłączy poprzez wykorzystywany tu protokół MQTT.

The screenshot shows a web-based interface for managing devices. At the top, there is a header bar with three tabs: "LabRoom-s1" (highlighted in blue), "devSensor" (highlighted in red), and "Device". Below the header, there is a navigation bar with five items: "Identity" (highlighted in blue), "Device Information", "Recent Events", "State", and "Logs".

The main content area displays the following device information:

Device ID	LabRoom-s1
Device Type	devSensor
Date Added	12 kwi 2019 12:35
Added By	[Redacted]
Connection Status	Disconnected

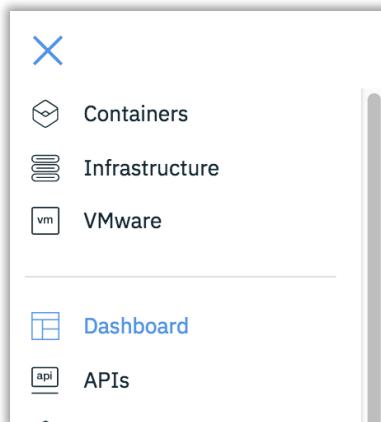
5. POWIĄZANIE USŁUG W IBM CLOUD

Usługi tworzone w ramach IBM Cloud mogą się wzajemnie autoryzować poprzez ręczne wpisanie danych dostępowych lub automatycznie, dzięki wzajemnym powiązaniom ustawianym w interfejsie IBM Cloud. Gdy aplikacje ze sobą powiążemy – to zostaną automatycznie ustawione odpowiednie zmienne środowiskowe pozwalające na podłączenie się do powiązanych usług).

Aby uprościć przebieg naszych warsztatów wykorzystamy tę drugą możliwość i połączymy instancje Node-RED oraz Watson IoT Platform.

57. W nowej zakładce przeglądarki wejdź na stronę bluemix.net i zaloguj się do swojego środowiska (ew. Użyj karty z już otwartym interfejsem IBM Cloud).

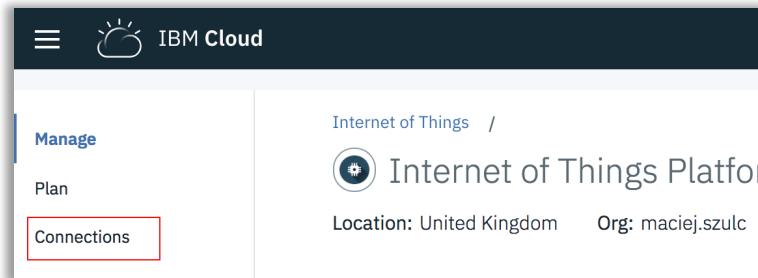
58. Za pomocą menu  przejdź do swojego ekranu Dashboard



59. Wybierz usługę „Internet Of Things Platform” poprzez kliknięcie na jej nazwie

Cloud Foundry Services 2/4 Used	
Name	Service Offering
Internet of Things Platform-e6	Internet of Things Plat...

60. Wybierz polecenie „Connections” z lewej strony ekranu



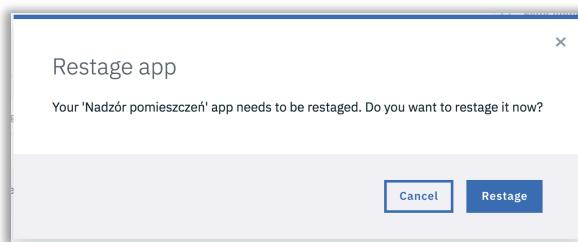
61. Wybierz polecenie „Create Connection”

Create connection +

62. Kliknij „Connect” przy swojej usłudze Node-Red.

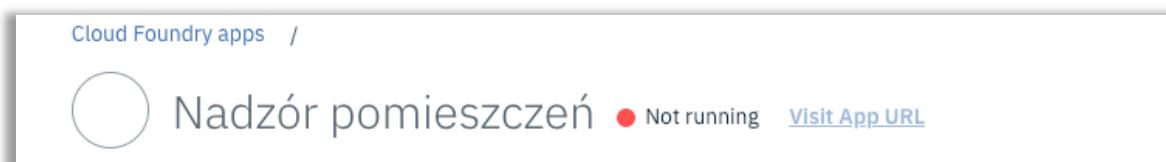
NAME	ROUTE	STATUS
Nadzór pomieszczeń	uw-maciej-18.mybluemix.net	Running

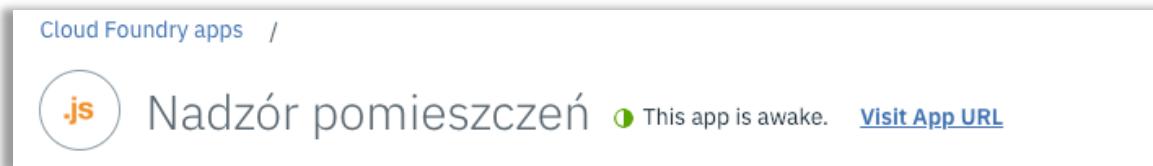
63. Zaakceptuj rekonfigurację i restart usługi Node-RED



64. Aby sprawdzić postęp zadania powróć do swojego ekranu Dashboard, wybierz usługę „Nadzór pomieszczeń” i poczekaj na jej uruchomienie. Operacja ta potrwa kilka minut (aplikacja jest na nowo instalowana z repozytorium git. Aby sprawdzić status możesz również odświeżyć stronę (choć status aplikacji powinien się zmienić i bez tej operacji)).

Zwróć uwagę, iż poprzez menu z lewej strony masz dostęp do informacji o powiązaniach (Connections) oraz logach Twojej aplikacji.

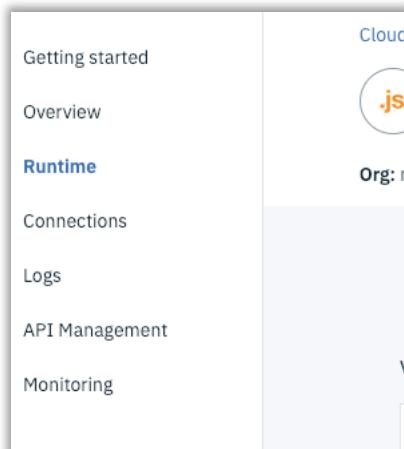




65. Kontynuuuj wykonywanie ćwiczenia dopiero gdy usługa wystartuje.

Uwaga: jeśli nie możesz powiązać usługi (nie pokazuje się Node-Red jako element możliwy do powiązania) – wówczas operację wykonaj „w drugą stronę”: wejdź do Node-Red i powiąż z Watson IOT.

66. Jeśli chcesz zobaczyć jaki jest efekt wykonanej operacji powiązywania to wejdź do menu „Runtime”



67. Następnie wybierz „Environment Variables”, gdzie możesz zobaczyć zawartość odpowiednich zmiennych środowiskowych VCAP_SERVICES. Będą one automatycznie używane przez powiązane ze sobą aplikacje.

Memory and instances Environment variables

VCAP_SERVICES

```
{  
  "iotf-service": [  
    {  
      "name": "Internet of Things Platform-l3",  
      "instance_name": "Internet of Things Platform-l3",  
      "binding_name": null,  
      "credentials": {  
        "iotCredentialsIdentifier": "a2g6k39sl6r5",  
        "mqtt_host": "kmymla.messaging.internetofthings.ibmcloud.com",  
        "mqtt_u_port": 1883,  
        "mqtt_s_port": 8883,  
        "http_host": "kmymla.internetofthings.ibmcloud.com",  
        "org": "kmymla",  
        "apiKey": "a-kmymla-6hhtr01ei",  
        "apiToken": "c(HCjGdw@+&?OTA-NS",  
        "error": null  
      },  
      "syslog_drain_url": null,  
      "volume_mounts": [],  
      "label": "iotf-service",  
      "provider": null,  
      "plan": "iotf-service-free",  
      "tags": [  
        "internet_of_things",  
        "iotf-service"  
      ]  
    }  
  ]  
}
```

6. WYŁĄCZENIE WYMOGÓW WZGLEDEM SZYFROWANIA POŁĄCZEŃ

Domyślna konfiguracja usługi Watson IOT wymusza stosowanie połączeń szyfrowanych. Na potrzeby naszego ćwiczenia pozwolimy jednak na zmniejszenie wymaganego poziomu zabezpieczeń i logowanie się otwartym połączeniem, bez dodatkowego szyfrowania.

Oczywiście w przypadku produkcyjnych instalacji należy przemyśleć wpływ takiej decyzji na bezpieczeństwo środowiska!

68. Powróć do głównego ekranu Watson IOT Platform

The screenshot shows the main interface of the IBM Watson IoT Platform. On the left is a vertical sidebar with icons for different sections: All Devices (selected), Device Types, Device ID, Identity, Device Information, Recent, and Security. The main content area has a header "IBM Watson IoT Platform" and tabs for "Browse", "Action", and "Device Types". Below this is a sub-header "All Devices" and a "Diagnose" button. A message states: "This table shows a summary of all devices that have been using different criteria. To get started, you can add device". There is a search bar with "Device ID" and a dropdown arrow. A table lists one device: "LabRoom-s1" (Device ID: LabRoom-s1, Device Type: devSensor). Below the table are three tabs: "Identity" (selected), "Device Information", and "Recent".

69. Wybierz ikonkę „Security”



70. Wybierz ikonkę ołówka obok „Connection Security”

Connection Security

Configure the security level for device connection.



71. Zmień „Default Rule” tak, aby w polu „Security Level” pojawił się „TLS Optional”

Scope	Security Level	Predicted Compliance ⓘ	# of Devices
Default	TLS Optional	↻ Refresh compliance	1 device

72. Kliknij „Save”

Cancel Save

SYMULATOR URZĄDZENIA

Nasz symulator będzie składał się z następujących węzłów node-red:

- **Inject (grupa: input; pojawia się jako „timestamp”)** – zapewniający cykliczne lub na żądanie wysyłanie próbki danych
- **Function (grupa: function)** – przygotowujący odpowiednią strukturę danych
- **Mqtt (grupa output)** – wysyłający dane do platformy MQTT
- **Change (grupa function, pojawia się jako set msg.payload)** – ustawia dowolne pole w bieżącym przepływie lub w zmiennych globalnych
- **Debug (grupa output; pojawia się jako msg.payload)** – pozwalająca na weryfikację treści wysyłanych komunikatów i debugowanie programu

Symulator będzie wysyłał do platformy IoT komunikat o formacie:

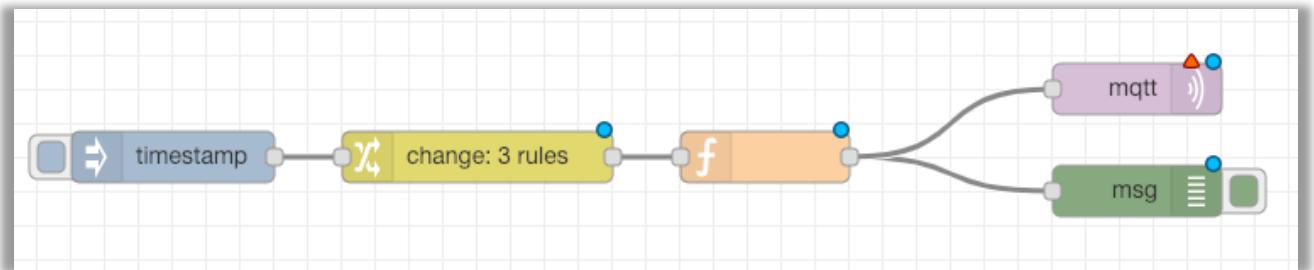
```
{
  "topic": "iot-2/type/devSensor/id/LabRoom-s1/evt/status/fmt/json",
  "payload": {
    "d": {
      "timestamp": 1555323411534,
      "temp": 26.66,
      "humi": 87.55,
      "doorA": 2,
      "doorB": 2,
      "doorC": 2
    }
  },
  "deviceId": "LabRoom-s1",
  "deviceType": "devSensor",
  "eventType": "status",
  "format": "json",
  "_msgid": "db490426.6cecb8"
}
```

7. UTWORZENIE SYMULATORA IOT

73. Przejdz do swojego edytora Node-RED: skorzystaj z uprzednio otwartej zakładki lub na ekranie Dashboard kliknij na polu „Route” Twojej usługi Node-Red.
Otworzy się nowa zakładka z dostępem do edytora, kliknik „Go to Node-RED flow editor”

Cloud Foundry Apps 256 MB/1 GB Used				
Name	Route	Memory (MB)	State	
Nadzór pomieszczeń	pwr-maciej-1.eu-gb...	256	Running (1/1)	⋮

74. W zakładce z programem „Symulator” którą uprzednio stworzyliśmy dodaj do obszaru roboczego wymienione powyżej węzły i połącz je ze sobą jak poniżej:

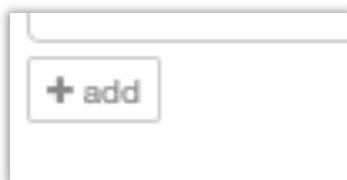


75. Węzeł Change wykorzystamy do ustawienia pola deviceId, deviceType i topic w obiekcie msg który jest przesyłany pomiędzy poszczególnymi węzłami programu.
Pole topic jest ustawiane zgodnie z regułami obowiązującymi w Watson IOT Platform – należy je wpisać dokładnie jak poniżej (iot-2/evt/status/fmt/json)



Wpisane wartości deviceId, deviceType muszą być zgodne z polem Device Id / Device Type dla urządzenia, które zostało utworzone w kroku 0 ze strony 16

Kliknij na przycisku „Add” węzła change dwa razy aby dodać drugą i trzecią regułę

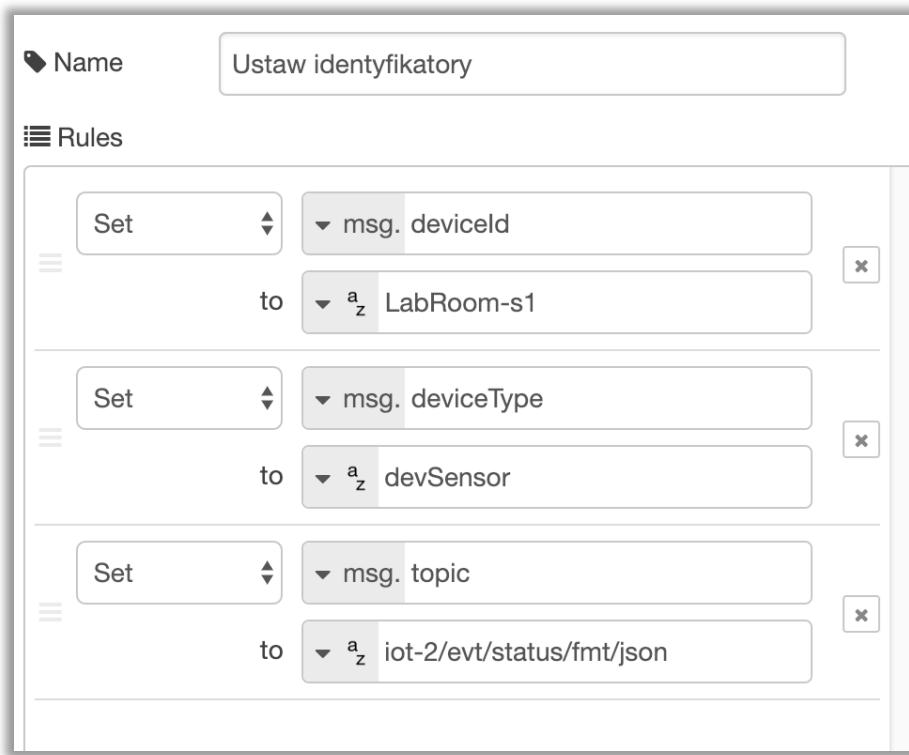


76. A następnie skonfiguruj jak poniżej

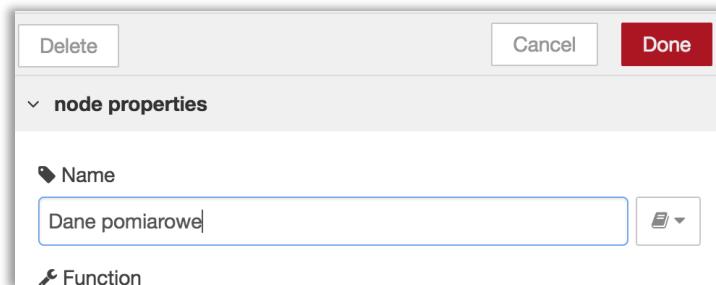


dostosowując zawartość zmiennych do tego co uprzednio ustawiliś w usłudze IoT Platform





77. Kolejny krok to skonfigurowanie procedury wysyłającej dane do platformy MQTT. W tym celu dwukrotnie kliknij na żółty węzeł function.
Nadaj mu nazwę.



78. W polu „Function” napisz program wysyłający dane pomiarowe. Możesz użyć gotowego kodu przygotowanego na potrzeby niniejszego ćwiczenia (UWAGA NA ZBĘDNE SPACJE!!!):

```
function round(num) {
    return +(Math.round(num + "e+2") + "e-2");
}

var oldTemp = context.get('temp')||20;
var oldHumi = context.get('humi')||50;
var doorA = flow.get('doorA')||0;
var doorB = flow.get('doorB')||0;
var doorC = flow.get('doorC')||0;

var currDate = new Date();
var myTime=currDate.getTime();

var currTemp=round(Math.random()+oldTemp-0.5);
var currHumi=round(Math.random()+oldHumi-0.5);

if (currHumi>90) currHumi=90;
if (currHumi<10) currHumi=10;
if (currTemp>30) currTemp=30;
if (currTemp<15) currTemp=15;

context.set('temp', currTemp);
context.set('humi', currHumi);
var status={"d":
{
    "timestamp" : myTime,
    "temp" : currTemp,
    "humi" : currHumi,
    "doorA": doorA,
    "doorB": doorB,
    "doorC": doorC
}
}

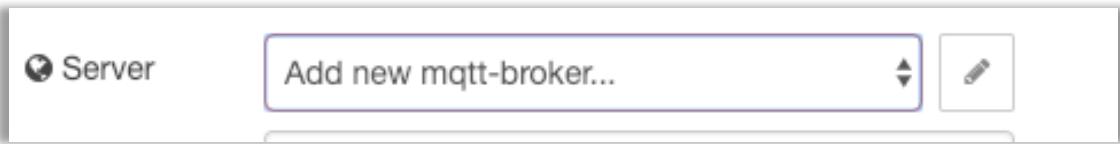
msg.payload=status;
return msg;
```

79. Wciśnij „Done” i zamknij węzeł function

80. Kliknij dwukrotnie na węźle „MQTT out”



81. Kliknij ikonkę ołówka przy „Add new mqtt-broker”



Pole „server” wypełnij jako

<TwójOrganizationID>.messaging.internetofthings.ibmcloud.com



Wpisane tu wartości muszą być zgodne z polem „Organization ID” dla urządzenia, które zostało utworzone w kroku 0 ze strony 16



np. *kmyla.messaging.internetofthings.ibmcloud.com*

82. Pole Client ID wypełnij zgodnie ze schematem:

d:{organization_ID}:{device_type}:{device_ID}



Wpisane tu wartości muszą być zgodne z polem „Organization ID” dla urządzenia, które zostało utworzone w kroku 0 ze strony 16

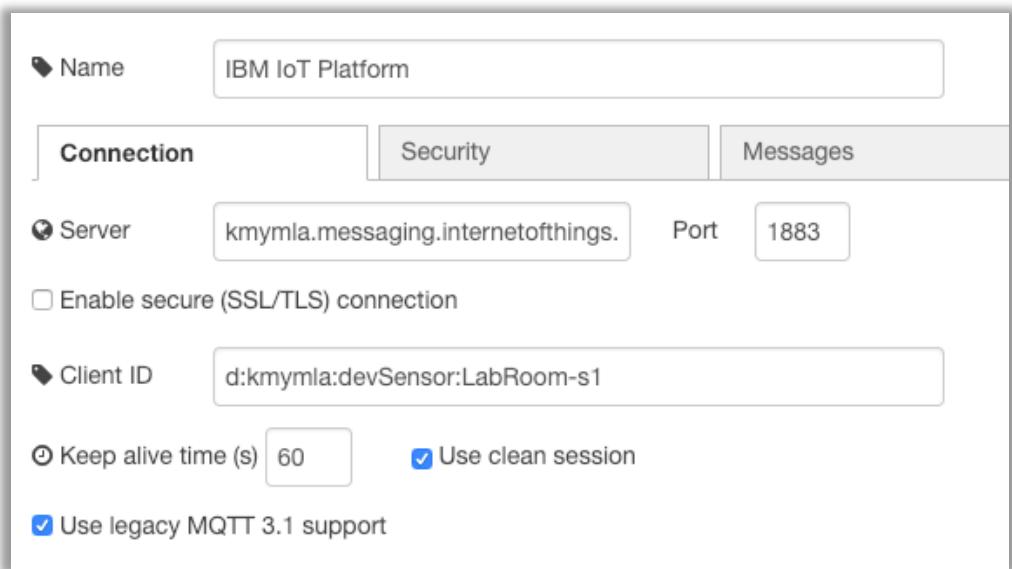


np.: *d:kmyla:devSensor:LabRoom-s1*

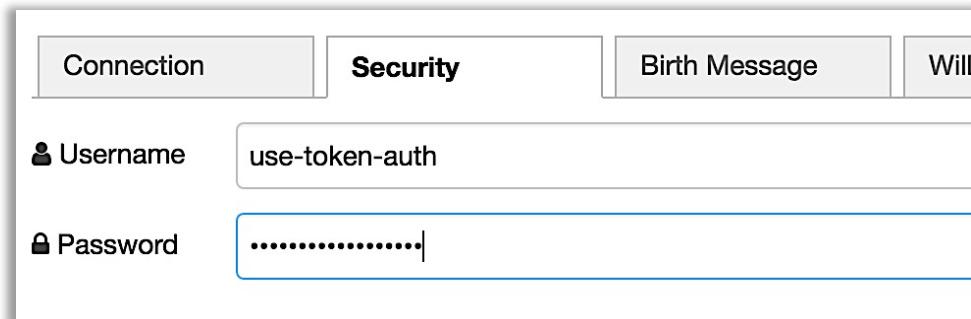
Oczywiście powyższy przykład musisz zmienić na swój własny identyfikator, typ i nazwę urządzenia.

Uwaga: podczas kopирования z plików PDF często zdarzają się błędy (np. zbędne znaki spacji). Sprawdź dokładnie wszystkie wpisywane informacje!

Uwaga na literkę d: na początku! Jest ona niezbędna!!!

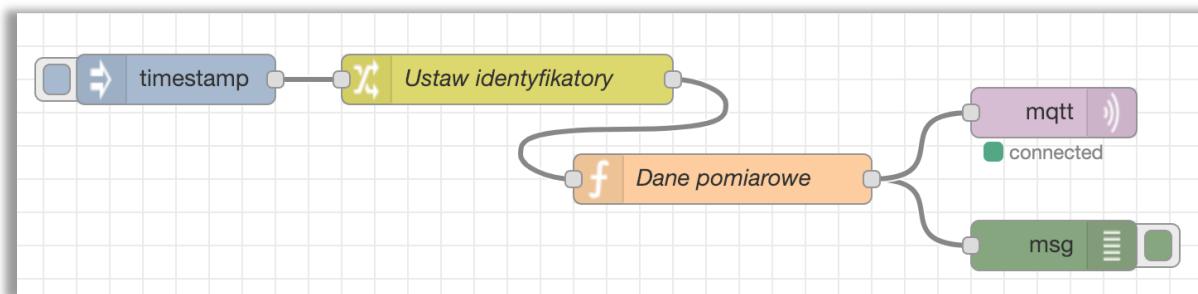


83. W zakładce "Security" pole "Username" wypełnij jako "use-token-auth" a w pole "Password" wpisz swój utworzony w kroku 0 ze strony 16 token.



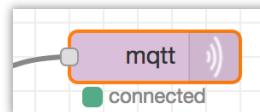
84. Kliknij przycisk "Add" aby zapisać zmiany

85. Kliknij przycisk "Done" aby zakończyć edycję. Program powinien wyglądać jak poniżej:



86. Uruchom program przyciskiem „Deploy” (prawy górny róg)

87.  Poczekaj, aż status operacji "Publish to IoT" zmieni się na „connected”.



Bardzo często zdarza się, iż połączenie nie jest nawiązywane – zawsze oznacza to, iż jedno z pól nie zostało poprawnie wypełnione.

Sprawdź zatem:

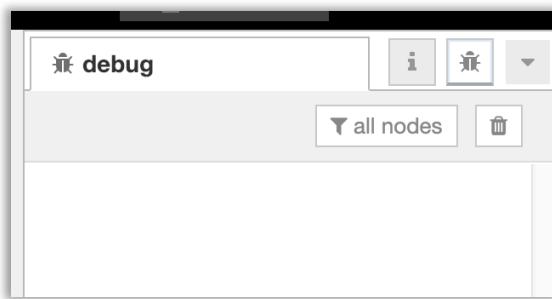
-  **Adres serwera:** musi być w formacie
`<twoja organizacja>.messaging.internetofthings.ibmcloud.com`
Pole `<twoja organizacja>` znajdziesz również panelu „IoT Platform” w prawej górnej części ekranu:

-  **Pole Client ID** musi być w formacie:
`d:{organization_ID}:{device_type}:{device_ID}`
`{device_type}` i `{device_ID}` znajdziesz np. w zakładce „Devices” Twojej usługi Watson IoT Platform:

Device ID	Device Type	Class ID	Date Added
1 result			
Bud1Pok312A	devSensor	Device	14 mar 2018 13:22

-  **Pole Username** musi zawierać tekst „use-token-auth”
-  **Pole Password** musi zawierać token który zapisałeś podczas tworzenia urządzenia IoT.
- **Sprawdź dokładnie, czy w poszczególnych polach w wyniku kopiowania nie pojawiły się zbędne spacje!**

88. Jeśli wszystko działa prawidłowo (**MQTT MA STATUS CONNECTED**) – przełącz widok panelu bocznego na „debug”:



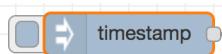
89. Uruchom przepływ „poduszką” na węźle Inject i sprawdź, czy nie pojawiają się błędy:

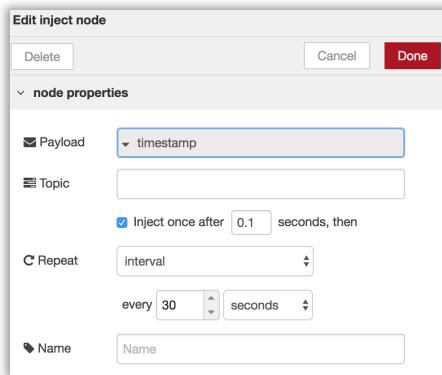


90. Prawidłowy wygląd okna powinien być jak poniżej (rozwiń obiekt):

```
12.04.2019, 16:22:29  node: 59a12910.092c78
iot-2/evt/status/fmt/json : msg : Object
  ▼object
    _msgid: "6ade0f87.57734"
    topic: "iot-
2/evt/status/fmt/json"
  ▼payload: object
    ▼d: object
      timestamp: 1555078949054
      temp: 19.53
      humi: 50.38
      doorA: 0
      doorB: 0
      doorC: 0
    deviceId: "LabRoom-s1"
    deviceType: "devSensor"
```

91. Jeśli wszystko działa prawidłowo (**MQTT MA STATUS CONNECTED**) – zmieniamy węzeł Inject

(Timestamp)  tak, aby komunikaty były wysyłane cyklicznie:



92. Uruchom kod poprzez wciśnięcie „Deploy”



8. WERYFIKACJA KOMUNIKACJI

93. Wchodzimy ponownie do zakładki z naszą usługą IBM Watson IoT Platform – poprzez Dashboard lub dzięki wykorzystaniu otwartej uprzednio zakładki.



94. Korzystając z lewego paska wybierz menu „Devices”

95. Klikamy na naszym urządzeniu

96. Sprawdzamy „Connection Status” (musi być w stanie “Connected”).



Jeśli urządzenie się nie podłączyło – musisz sprawdzić poprzednie kroki. **Do czasu uzyskania poprawnej komunikacji nie wykonuj dalszych kroków ćwiczenia. Sprawdź wówczas symulator zaimplementowany w Node-RED.**

A screenshot of the IBM Watson IoT Platform interface showing the details for a device named "LabRoom-s1" of type "devSensor". The "Identity" tab is selected, displaying the following information:

Device ID	LabRoom-s1
Device Type	devSensor
Date Added	12 kwi 2019 12:35
Added By	mszulc@gmail.com
Connection Status	Connected
Connection Time	12 kwi 2019 16:24
Client Address	169.46.115.121 (Insecure)

97. Czekamy minutę lub dwie, aby usługa symulatora IoT wysłała kilka próbek pomiarowych.

98. Weryfikujemy czy w „Recent Events” na karcie naszego urządzenia pojawiają się kolejne zdarzenia:

The screenshot shows the 'Recent Events' tab for a device named 'LabRoom-s1'. The table displays five events, all labeled 'status', showing JSON data with timestamps and temperatures. The last event was received 'kilka sekund temu' (several seconds ago).

Event	Value	Format	Last Received
status	{"d":{"timestamp":1555080063174,"temp":22.1}}	json	kilka sekund temu
status	{"d":{"timestamp":1555080055236,"temp":22.1}}	json	kilka sekund temu
status	{"d":{"timestamp":1555080055094,"temp":22.1}}	json	kilka sekund temu
status	{"d":{"timestamp":1555080054930,"temp":22.1}}	json	kilka sekund temu
status	{"d":{"timestamp":1555080054785,"temp":21.9}}	json	kilka sekund temu

99. Weryfikujemy czy w „State” na karcie naszego urządzenia pojawiają się kolejne dane:

The screenshot shows the 'State' tab for the same device. It displays a hierarchical structure under 'd' with properties like timestamp, temp, humi, doorA, doorB, and doorC, each with their corresponding values and types.

Property	Value	Type	Event
d		Object	status
timestamp	1555080041492	Number	status
temp	20.87	Number	status
humi	50.03	Number	status
doorA	0	Number	status
doorB	0	Number	status
doorC	0	Number	status

100. Udało nam się utworzyć instancję usługi IOT Platform, symulator urządzenia oraz spiąć całość razem. Gratulacje!

WIZUALIZACJA DANYCH POMIAROWYCH

10. UTWORZENIE PROSTEJ WIZUALIZACJI DANYCH W PLATFORMIE IOT

Usługa IoT Platform jest brokerem danych pomiarowych – swoistym urzędem pocztowym odbierającym i dostarczającym przesyłki. Zwykle nie służy on do ich wizualizacji (wyjąć go w tym zadaniu podłączone aplikacje) – tym niemniej posiada proste mechanizmy wizualizacji informacji. W kolejnym ćwiczeniu będziesz miał możliwość się z nimi zapoznać.

Sprawdzimy również, czy nasz symulator prawidłowo wysyła dane pomiarowe.

101. Powróć do głównego ekranu Watson IOT Platform

The screenshot shows the 'IBM Watson IoT Platform' interface. On the left is a vertical sidebar with icons for different sections: 'Browse' (selected), 'Action', 'Device Types', 'Boards' (highlighted in blue), 'Diagnose', 'Logs', 'Metrics', 'Analytics', and 'Settings'. The main area is titled 'Browse Devices' and contains a summary message: 'This table shows a summary of all devices that have been added. It can be filtered, organized, and visualized using different criteria. To get started, you can add devices by using the Add Device button, or by importing a CSV file.' Below this is a table with columns: 'Device ID' (dropdown), 'Device Type' (dropdown), 'Class ID' (dropdown), and 'Actions'. One row is visible: 'LabRoom-s1' (Device ID), 'devSensor' (Device Type), 'Device Class 1' (Class ID), and 'Actions' (button).

102. Wybierz ikonkę „Boards”



103. Wybierz „Create New Board”



104. Wypełnij nazwę i opis swojej tablicy, kliknij „Next”

Board settings

Provide a name and description for your new board.

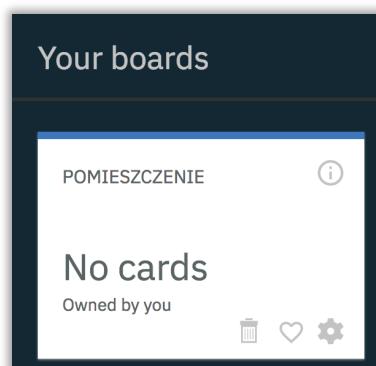
Board name
Pomieszczenie

Description
Dane z czujników pomiarowych

Make this board my landing page.
 Favorite (this also adds this board to your navbar)

105. Na zakładce „Board Settings” kliknij „Submit”

106. Na ekranie powinna się pojawić nowa tablica



107. Kliknij na niej 2 razy aby ją otworzyć

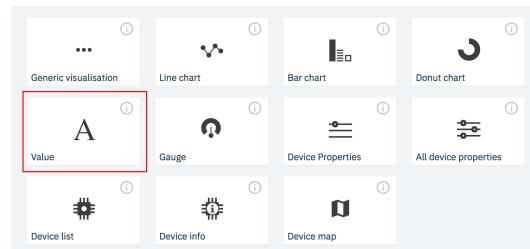
108. Kliknij „Add New Card”

109. Jako typ wizualizacji wybierz „Value” z grupy „Devices”

Create Card

Card type
Select card type

Devices



110. Wybierz czujnik, który dodałeś i kliknij „Next”

Create Value Card

Specify the data source for the card

Devices

Search for card data sources using the filter:

Device ID Device Type

<input checked="" type="checkbox"/> LabRoom-s1	devSensor
--	-----------

⚠ Jeśli czujnik nie pojawia się – sprawdź czy na pewno Twój symulator podłączył się do usługi IoT – krok 96, ew. zamknij i otwórz ponownie przeglądarkę

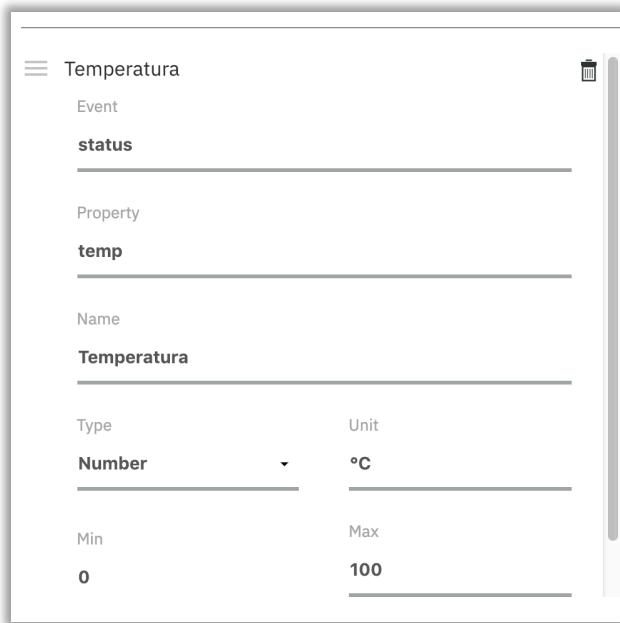
111. Kliknij „Connect new data set”

Connect new data set

112. Wybierz:

- Zdarzenie: “status”,
- property: “temp”,
- name: wpisz “Temperatura”,

- type: number,
- min: 0,
- max: 100,
- Unit: „ $^{\circ}\text{C}$ ”



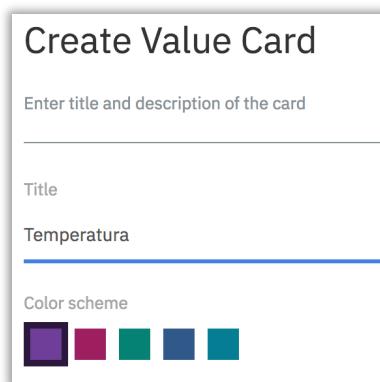
113. Kliknij next

114. Rozmiar: XS, kliknij “Next”

115. Title: wpisz “Temperatura”.

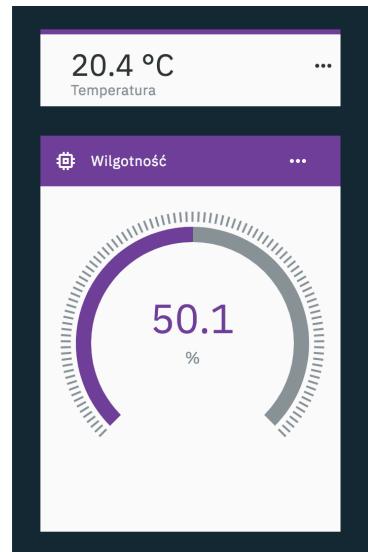
Schemat kolorów: do wyboru.

Zatwierdź poprzez „Submit”



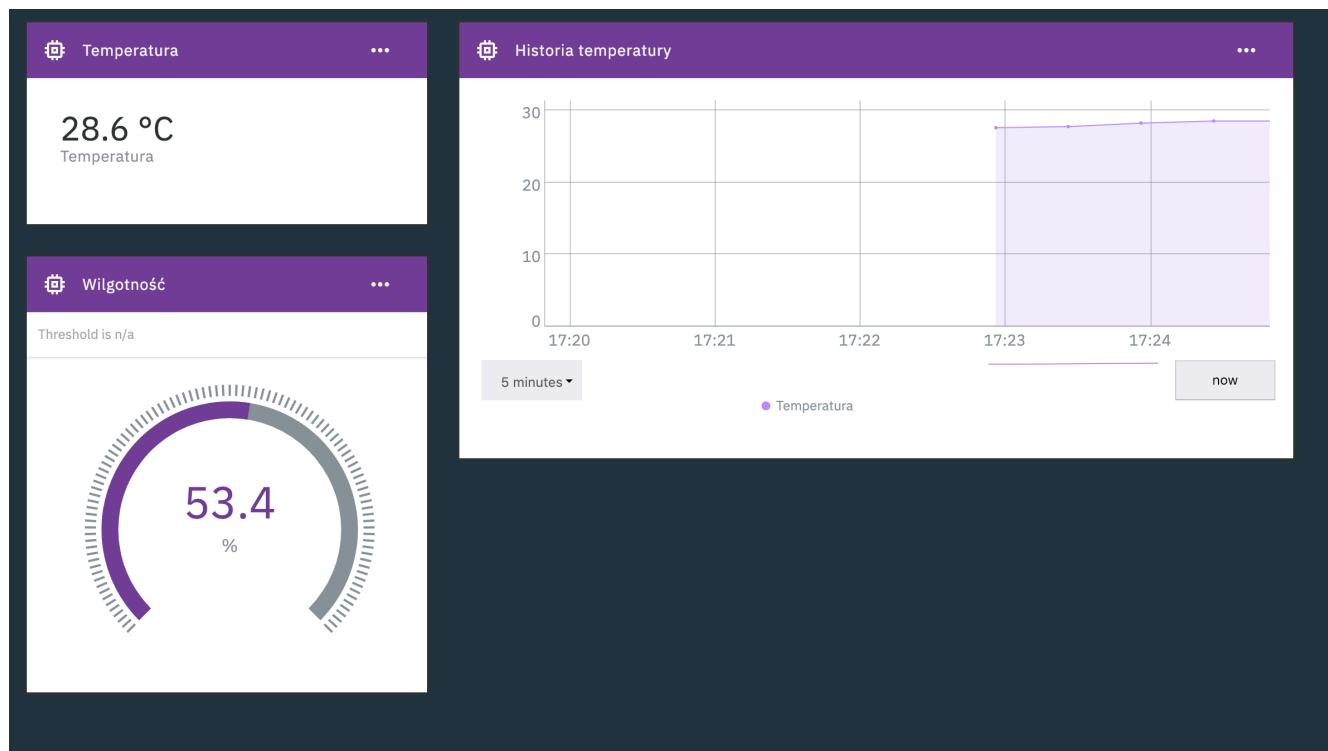
116. Zobacz, jak dane wysyłane przez nasz symulator wpływają na zawartość wykresu (możesz wygenerować dodatkowe „poduszka” węzła Inject w Node-RED).

117. Teraz dodamy widget z wilgotnością – dodaj widget typu gauge rozmiar „S” przedstawiający wartość „humi”.



118. Poprzez dodanie widgetu „Line chart” przedstawiającego pole „status.temp” dodaj prezentację historii zmian temperatury.

119. Uporządkuj widgety przesuwając je na ekranie



11. DODANIE SKŁADOWANIA DANYCH HISTORYCZNYCH

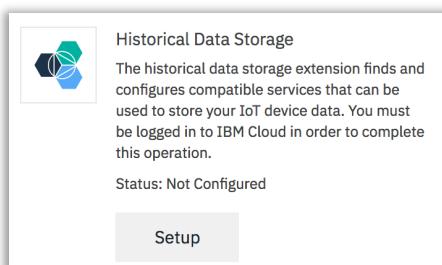
Karta, którą przygotowałeś prezentuje dane bieżące – po wyjściu i powrocie wartości historyczne się nie pojawią. Spróbujemy zatem temu zaradzić.

120. Dzięki funkcjonalności rozwiązań chmurowych dodanie składowania danych do późniejszego wykorzystania jest niezwykle proste – dlatego też skonfigurujemy i ten aspekt działania platformy Watson IOT.

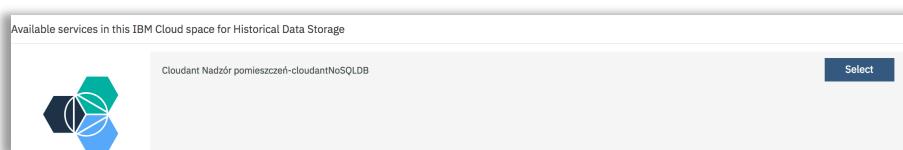
Wybierz zakładkę „Extensions”



121. Kliknij „Setup” przy „Historical Data Storage”



122. Wybierz instancję bazy danych Cloudant NoSQL poprzez przycisk „Select”



123. Pozostaw domyślne parametry i zatwierdź utworzenie połączenia przyciskiem „Done”

124. Zaakceptuj autoryzację w okienku poniżej.

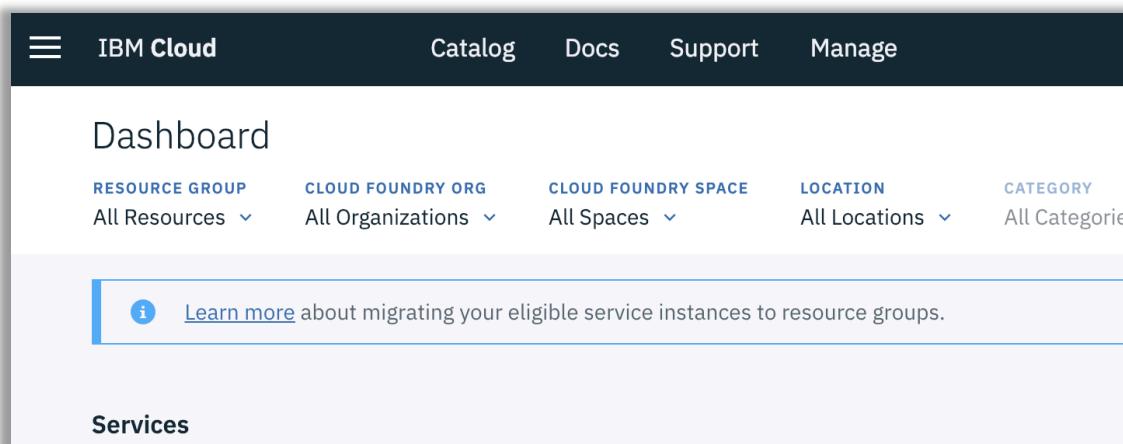
Jeśli okienko się nie pojawiło – sprawdź ustawienia „wyskakujących okienek” w przeglądarce.



Od tej pory dane pomiarowe są składowane w bazie informacji historycznych i prezentowane w wizualizacji Watson IOT Platform.

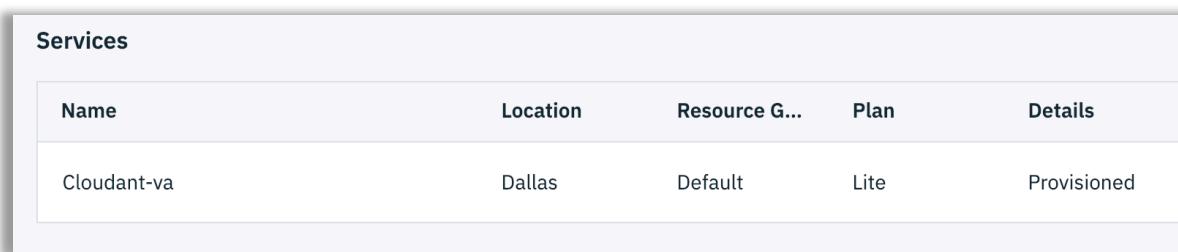
12. SPRAWDZENIE ZAWARTOŚCI BAZY DANYCH PRZECHOWUJĄCEJ DANE HISTORYCZNE.

125. Sprawdźmy, czy dane i jak są składowane informacje w bazie Cloudant. Przejdz do zakładki z głównym interfejsem IBM Cloud i wybierz „dashboard”



The screenshot shows the IBM Cloud dashboard. At the top, there's a navigation bar with 'IBM Cloud' and links for Catalog, Docs, Support, and Manage. Below the navigation bar, the word 'Dashboard' is displayed. Underneath 'Dashboard', there are several filter options: 'RESOURCE GROUP' (All Resources), 'CLOUD FOUNDRY ORG' (All Organizations), 'CLOUD FOUNDRY SPACE' (All Spaces), 'LOCATION' (All Locations), and 'CATEGORY' (All Categories). A callout box with an info icon contains the text: 'Learn more about migrating your eligible service instances to resource groups.' In the main content area, there's a section titled 'Services' which lists one service: 'Cloudant-va' located in 'Dallas' with a 'Default' resource group, 'Lite' plan, and 'Provisioned' status.

126. Wybierz usługę Cloudant z sekcji „Services” (**nie wybieraj z „Cloud Foundry Services”**)



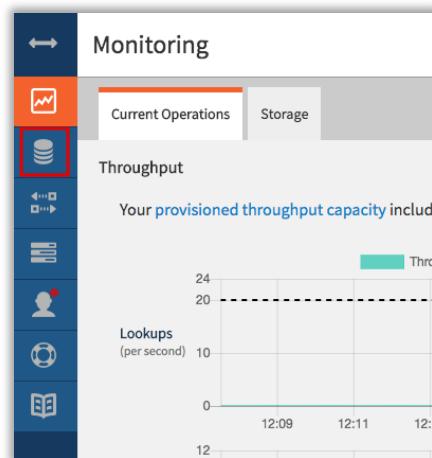
This screenshot shows the 'Services' section of the IBM Cloud dashboard. It displays a table with columns: Name, Location, Resource G..., Plan, and Details. There is one entry: 'Cloudant-va' located in 'Dallas' with a 'Default' resource group, 'Lite' plan, and 'Provisioned' status.

127. Kliknij „Launch”



This screenshot shows the details for the 'Cloudant-va' service. At the top, it shows the service name 'Cloudant-va' with a Cloudant icon. Below that, it says 'Resource Group: Default' and 'Location:'. The main description reads 'Cloudant NoSQL DB'. On the right side, there is a prominent green 'LAUNCH' button with a white arrow icon.

128. Wybierz ikonkę baz danych



129. Sprawdź, czy są bazy danych o nazwie rozpoczynającej się od „iotp”. Kliknij na tej, która ma najwięcej dokumentów.

Name	Size	# of Doc
iotp_kmymla_05453a16-de5d-4560-a295-72afcb717d8d_configuration	130.8 KB	0
iotp_kmymla_default_2019-04-12	8.7 KB	2
iotp_kmymla_default_2019-04-13	7.3 KB	1
iotp_kmymla_default_configuration	130.8 KB	0
nodered	19.9 KB	4

130. Sprawdź zawartość jednego z nich.

The screenshot shows a modal window for editing a document named 'iotp_bntnhp_default_2018-03-21'. The window has a 'Save Changes' button and a 'Cancel' button. The document content is a JSON object:

```

1 + [
2   "_id": "0011bc10-2c9d-11e8-a629-6d2ef6b65e33",
3   "_rev": "1-c375c0d5fb38fab8a784b4efe57c40c1",
4   "deviceType": "devSensor",
5   "deviceId": "Bud1Pok312A",
6   "eventType": "status",
7   "format": "json",
8   "timestamp": "2018-03-21T00:15:45.745Z",
9   "data": {
10     "d": {
11       "deviceId": "Bud1Pok312A",
12       "timestamp": 1521591345744,
13       "temp": 20.25,
14       "humi": 50.77,
15       "doorA": 1,
16       "doorB": 2,
17       "doorC": 2
18     }
19   }
20 ]

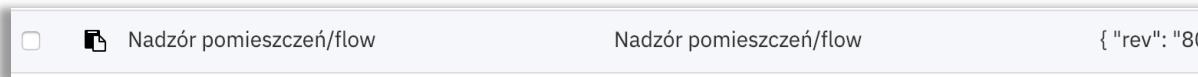
```

131. Wyjdź ponownie do ekranu prezentującego bazy danych.

132. Otwórz bazę „nodered”



133. Otwórz dokument „Nadzór pomieszczeń/flow”



```
nodered > Nadzór pomieszczeń/flow
Save Changes Cancel ↑

1 * [
  2   "_id": "Nadzór pomieszczeń/flow",
  3   "_rev": "49-222f472b1c2cb551657a5b4c30c8f2dc",
  4   "flow": [
  5     {
  6       "id": "9d8e91be.2a8ff",
  7       "type": "tab",
  8       "label": "Hello World",
  9       "disabled": false,
 10       "info": ""
 11     },
 12     {
 13       "id": "209441ba.db2a1e",
 14       "type": "tab",
 15       "label": "Simulator",
 16       "disabled": false,
 17       "info": ""
 18     },
 19     {
 20       "id": "c415e447.d49678",
 21       "type": "mqtt-broker",
 22       "z": "",
 23       "name": "IBM IoT Platform",
 24       "broker": "knymla.messaging.internetofthings.ibmcloud.com",
 25       "port": 1883
  
```

WIZUALIZACJA DANYCH POMIAROWYCH W NODE-RED DASHBOARD

Wizualizacja, którą przygotowałaś w ramach „Boards” usługi „Watson IOT” nie jest zwykle używana jako ekran roboczy w rzeczywistych implementacjach systemów IoT. Bardzo często są tworzone (i osadzane w chmurze) dedykowane aplikacje napisane w różnych językach programowania, również chętnie wykorzystywane są gotowe narzędzia do ich tworzenia.

W ramach platformy Node-Red wykorzystamy projekt „Node-Red Dashboard”, pozwalający na łatwe budowanie ciekawych interfejsów, nie tylko dla rozwiązań IoT.

13. INSTALACJA PAKIETU „NODE-RED-DASHBOARD”

- Przejdz do zakładki z edytorem Node-RED lub na ekranie Dashboard kliknij na polu „Route” Twojej usługi Node-Red. Otworzy się nowa zakładka z dostępem do edytora

Cloud Foundry Apps 256 MB/1 GB Used			
Name	Route	Memory (MB)	State
Nadzór pomieszczeń	pwr-maciej-1.eu-gb....	256	● Running (1/1) ⋮

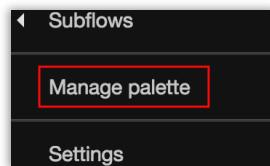
Otwórz edytor przyciskiem „Go to Node-RED flow editor”



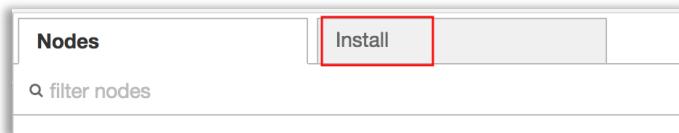
- Aby móc skorzystać z funkcji node-red dashboard do palety dostępnych elementów musimy dodać odpowiedni pakiet instalacyjny.
Otwórz menu obok przycisku „Deploy”



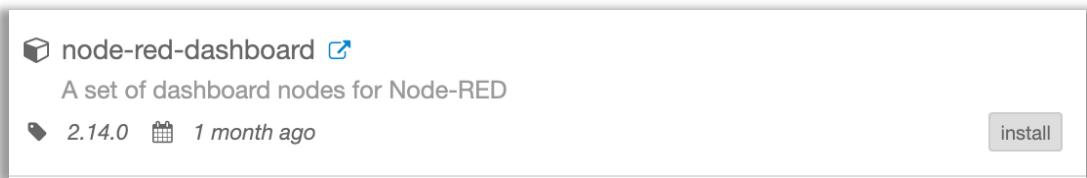
- Wybierz „Manage Palette”



138. Przejdź do zakładki „Install”



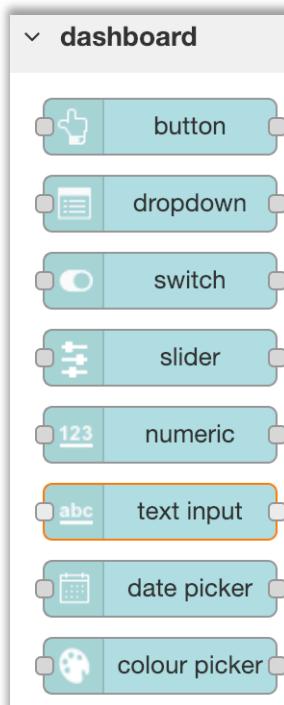
139. W polu wyszukiwania wpisz „node-red-dashboard”



140. Zainstaluj pakiet przyciskiem „Install”, ponownie potwierdź wybór „Install”.

141. Poczekaj na zakończenie instalacji, następnie zamknij „User settings” przyciskiem „Close”.

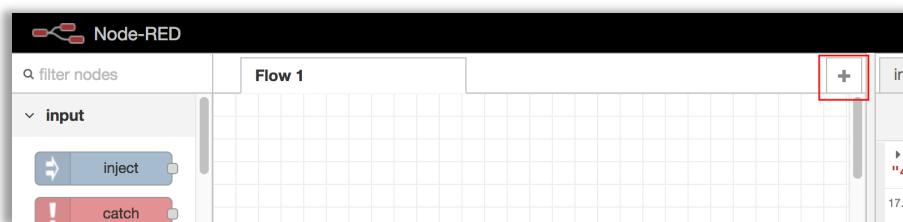
Nowe ikonki pojawiły się w przyborniku widgetów w kategorii „dashboard”



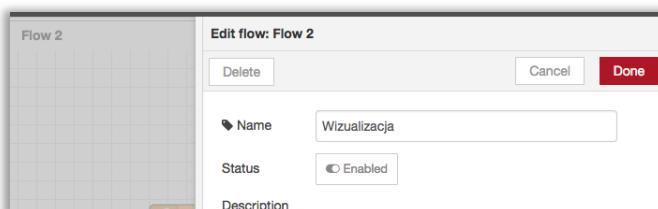
14. UTWORZENIE PODSTAWOWEJ WIZUALIZACJI

W tym ćwiczeniu utworzymy nowy program, który dołączy się do platformy Watson IOT i będzie z niej odbierał komunikaty które wysyłają urządzenia (zasubskrybuje wszystkie wiadomości od wszystkich urządzeń)

142. Do utworzenia naszej wizualizacji wykorzystamy nowy program (Flow). W tym celu na ekranie edytora kliknij „+” przy kartach Flow – utworzony zostanie nowy program o domyślnej nazwie „Flow 1”



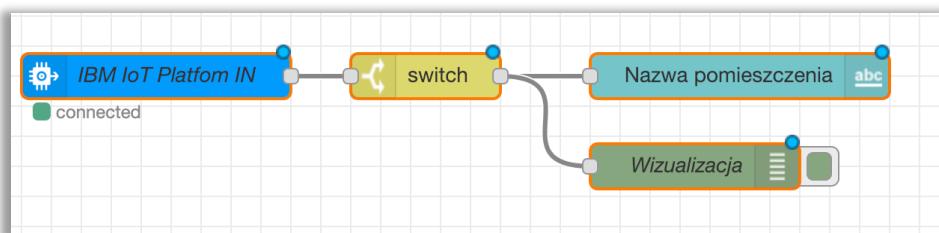
143. Zmieńmy nazwę obszaru roboczego Flow 1. W tym celu kliknij dwa razy na jego nazwie i ją zmień na „Wizualizacja”:



144. Do nowego ekranu roboczego „Wizualizacja” dodaj następujące węzły:

- **(input) ibmiot** – podłączenie do usługi Watson IOT wykorzystujące wewnętrzne mechanizmy autoryzacyjne.
- **(dashboard) text** – wyświetlający dowolny tekst na dashboardie node-red-dashboard
- **(output) debug**

Połącz je ze sobą jak poniżej



145. Skonfiguruj węzeł IBM IoT tak, jak przedstawiono poniżej:

The screenshot shows the 'Properties' dialog for an IBM IoT node. It contains the following settings:

- Authentication: Bluemix Service
- Input Type: Device Event
- Device Type: All or + (checkbox checked)
- Device Id: All or device id e.g. ab12cd231a21 (checkbox checked)
- Event: All or + (checkbox checked)
- Format: All or json (checkbox checked)
- QoS: 0
- Name: IBM IoT Platform IN

146. Wejdź do konfiguracji węzła (dashboard) text

147. Dodaj nową grupę przyciskiem ołówka obok pola „Add new ui_group”

The screenshot shows the 'Group' configuration dialog. It has a single input field: 'Add new ui_group...'.

148. Dodaj nową zakładkę dashboard'u „Pokój” w edytorze grup przyciskiem ołówka koło pola „Add new ui_tab”

The screenshot shows the 'Tab' configuration dialog. It has a single input field: 'Add new ui_tab...'.

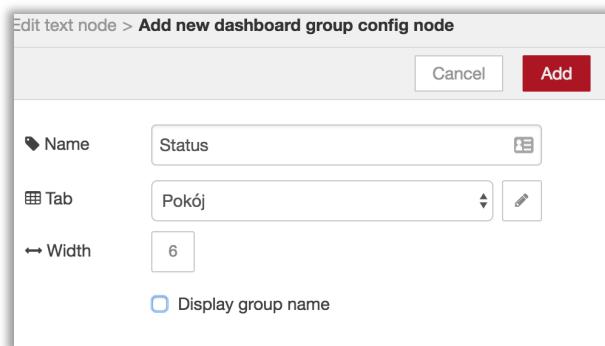
149. Nadaj nazwę zakładce i wybierz „Add”

The screenshot shows a modal dialog for creating a new tab. It has two fields:

- Name: Pokój
- Icon: dashboard

A red 'A' button is visible in the top right corner of the dialog.

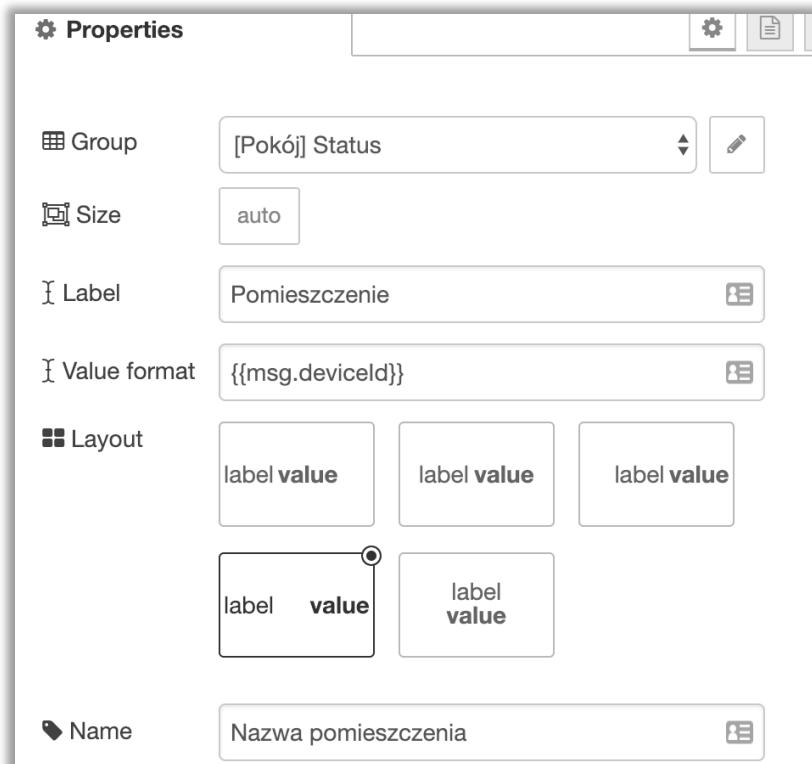
150. Nadaj nazwę grupie jak poniżej, wybierz „Add”



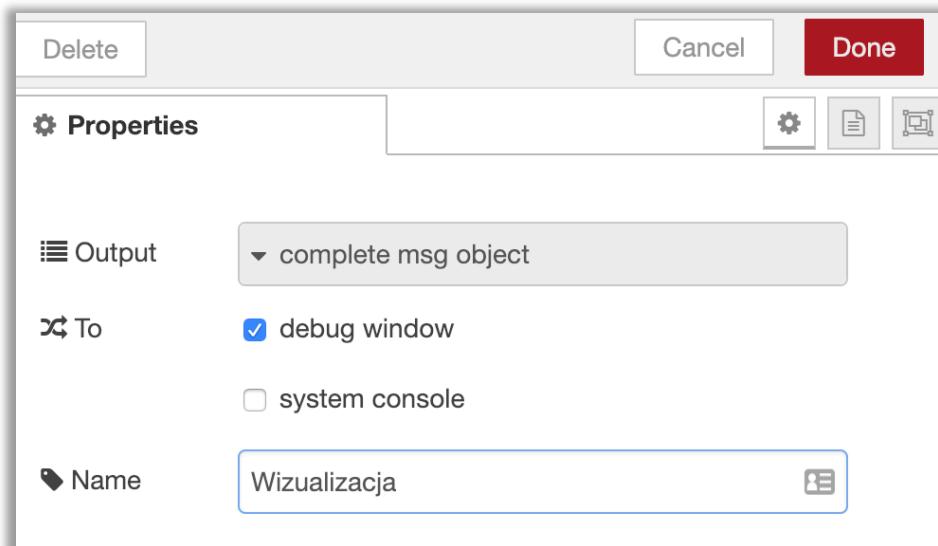
151. Dodaj

- nazwę pola (label)
- nazwę węzła (name)
- Value format jako {{msg.deviceId}}

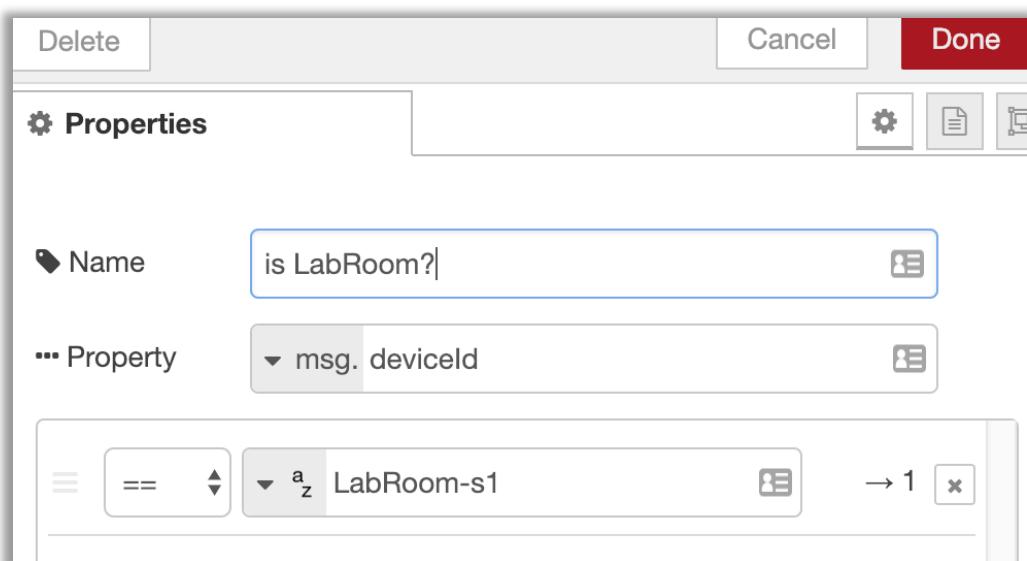
I pozostaw pozostałe wartości domyślnie i zatwierdź przez „Done”



152. Zmień konfigurację węzła „debug” tak, aby pokazywał cały obiekt a nie (jak to jest domyślnie) msg.payload



153. Wejdź do konfiguracji węzła „switch” i ustaw go tak, aby przepuszczał wyłącznie wiadomości od naszego symulowanego urządzenia:



154. Uruchom kod poprzez wciśnięcie „ Deploy”.

155. Wejdź do zakładki „debug” po prawej stronie, rozwiń obiekt „status” dla węzła debugowania „Wizualizacja” i sprawdź jego zawartość.

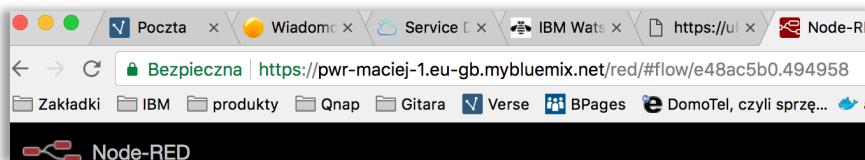
To właśnie na tych polach będziemy pracowali w dalszej części ćwiczenia.

```

12.04.2019, 17:59:47 node: Wizualizacja
iot-2/type/devSensor/id/LabRoom-s1/evt/status/fmt/json : msg : Object
  ▼ object
    topic: "iot-2/type/devSensor/id/LabRoom-s1/evt/status/fmt/json"
  ▼ payload: object
    ▶ d: object
      deviceId: "LabRoom-s1"
      deviceType: "devSensor"
      eventType: "status"
      format: "json"
      _msgid: "71f240d7.44066"

```

156. Interfejs node-red-dashboard jest dostępny pod tym samym adresem, co edytor Node-Red, ale pod nieco innym wywołaniem. Skopiuj URL edytora Node-Red



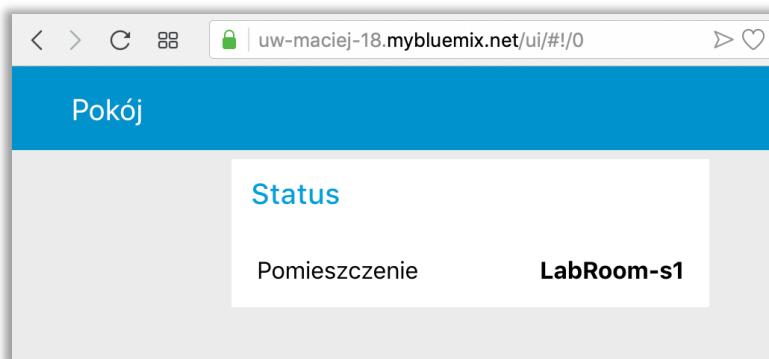
157. Otwórz nową kartę przeglądarki i wklej skopiowany adres, następnie go zmodyfikuj jak w poniższym przykładzie:

<https://uw-maciej-18.mybluemix.net/red/#flow/4d833349.162f2c>

zamieniamy na

<https://uw-maciej-18.mybluemix.net/ui>

Otwórz stronę o powyższym adresie



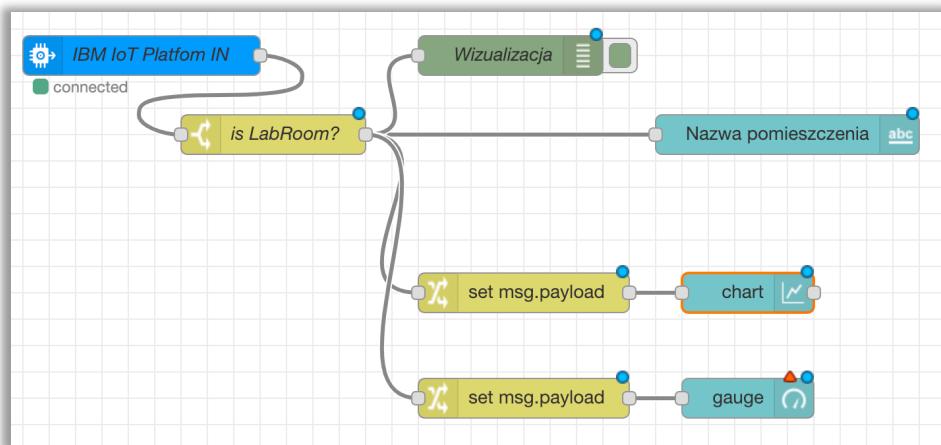
158. W kolejnym kroku dodamy kolumnę zawierającą dane o warunkach środowiskowych. W tym celu wykorzystamy nowe formy prezentacji danych pomiarowych: wskaźnik i wykres.

Widgety te wymagają specyficznej formy danych źródłowych.

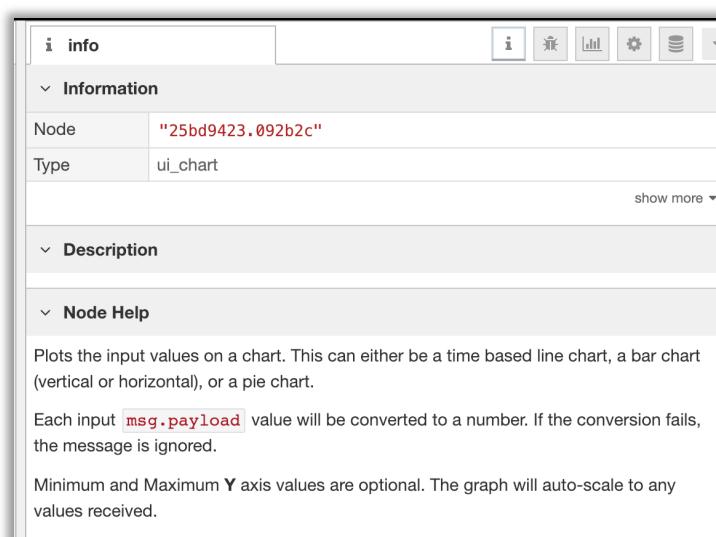
Do ekranu roboczego dodaj:

- „**gauge**” - rysuje wskaźnik na dashboardzie
- „**chart**” – rysuje wykres na dashboardzie
- dwa węzły „**set**” – zmieniają treść przesyłanego komunikatu

połącz je jak na rysunku poniżej:

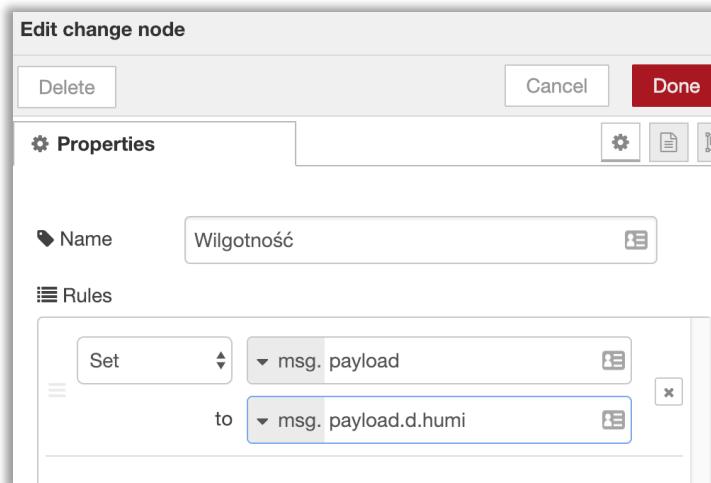


159. Kliknij na jedym z nowo dodanych widgetów, po czym w prawej kolumnie edytora przejdź do zakładki „info” i zapoznaj się z formatem danych które są oczekiwane przez daną formę wizualizacji. W ten sposób możesz sprawdzić parametry większości węzłów w node-Red.



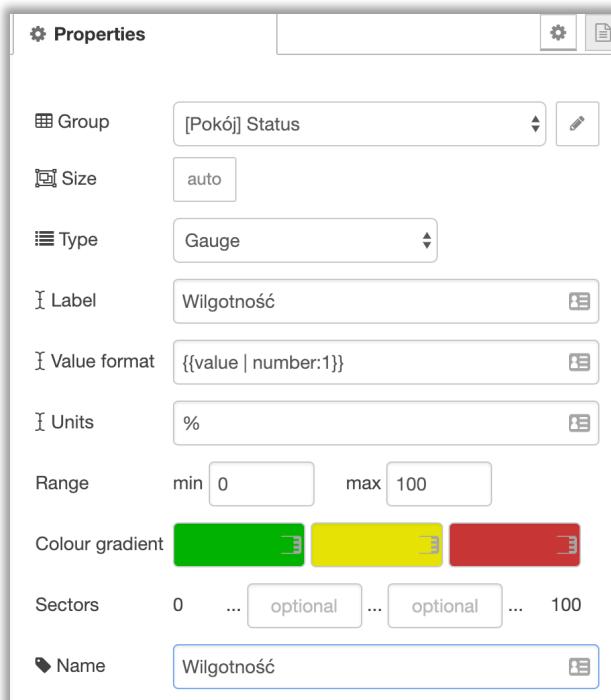
160. Kliknij dwa razy na węźle set połączonym z widgetem gauge.

Nadaj mu nazwę „**Wilgotność**” i skonfiguruj tak, aby do pola msg.payload kopiował zawartość msg.payload.d.humi:



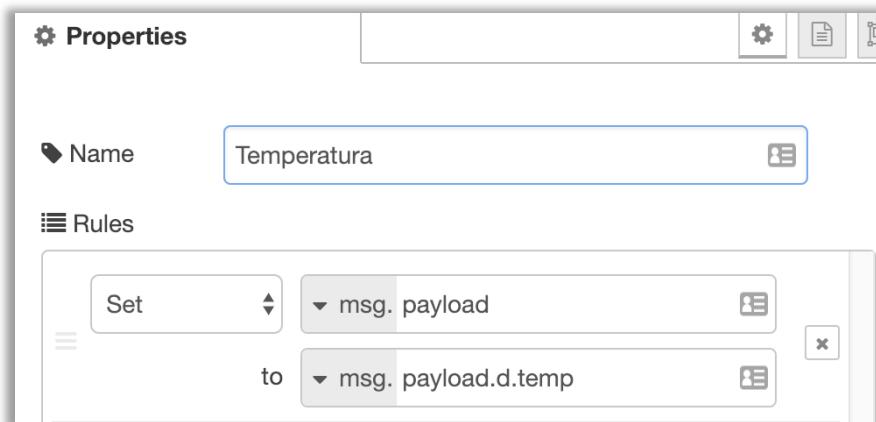
161. Kliknij dwa razy na węźle „gauge” i skonfiguruj go nadając:

- **Label:** „*Wilgotność*”
- **Value format:** „{{value | number:1}}”
- **Units:** „%”
- **Range:** od 0 do 100.
- **Group** „[Pokój] Status”



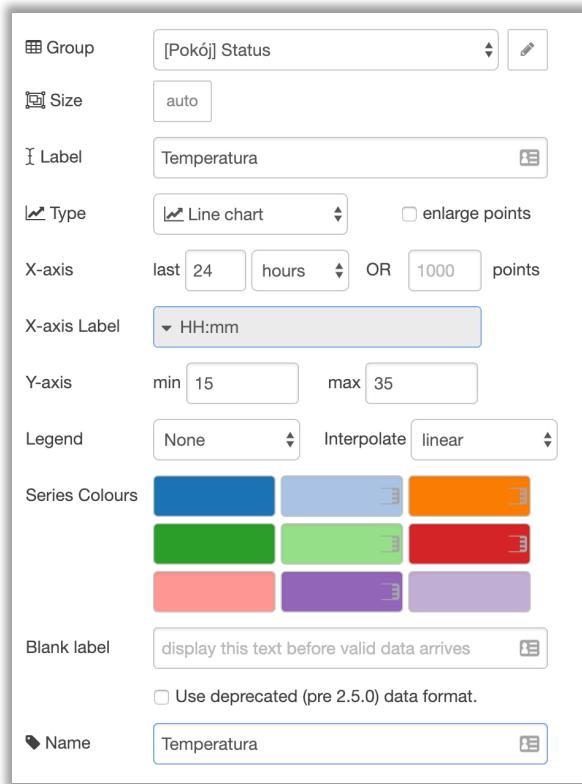
162. Kliknij dwa razy na węźle set połączonym z widgetem chart.

Nadaj mu nazwę „**Temperatura**” i skonfiguruj tak, aby do pola msg.payload kopiował zawartość msg.payload.d.temp:



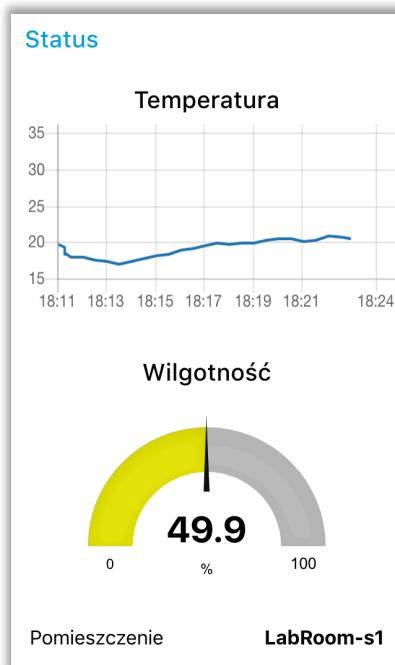
163. Kliknij dwa razy na węźle „chart” i skonfiguruj go nadając:

- **Label:** „Temperatura”
- **Type:** „Line chart”
- **X-Axis:** „last 24 hours”
- **X-Axis label:** „HH:mm”
- **Y-axis:** 15-35
- **Group:** „Status [Pokój]”

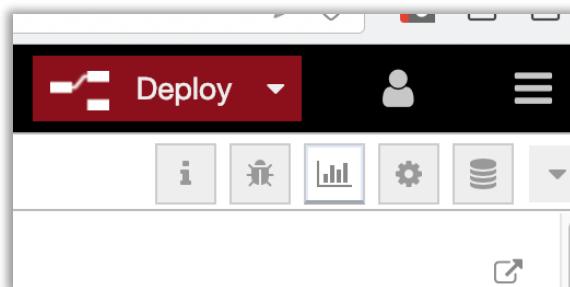


164. Zamknij okno edycji widgetu i uruchom program przyciskiem „Deploy”.

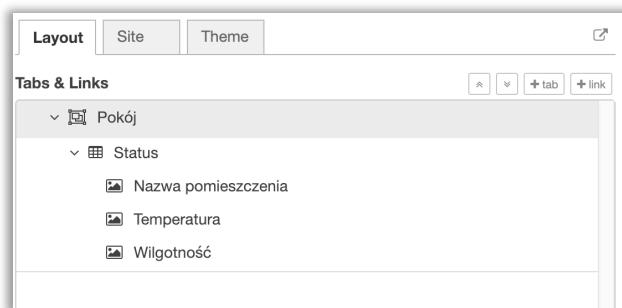
Przejdź do zakładki w której otworzyłeś widok budowanego interfejsu i sprawdź efekt.



165. Przeorganizujmy układ ekranu i jego parametry. W tym celu wejdź do konfiguracji dashboardu (na ekranie edytora Node-RED) – znajdziesz ją w prawym górnym rogu pod przyciskiem „Deploy”

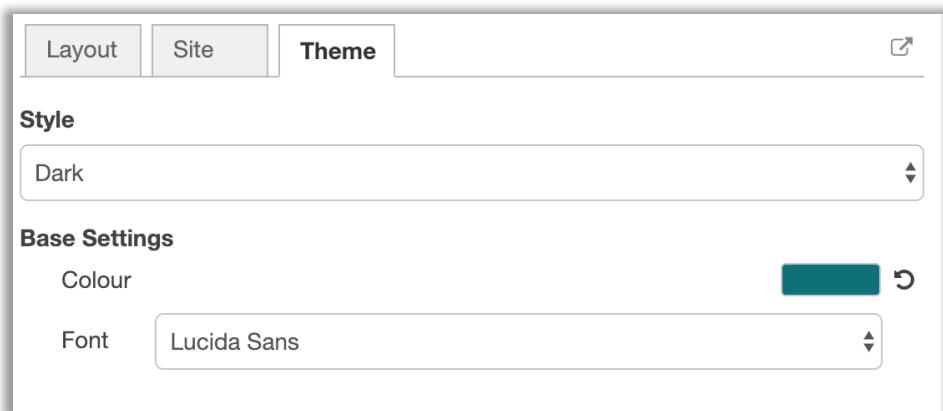


166. Rozwiń drzewko widgetów w zakładce „Layout” i uporządkuj je jak poniżej:



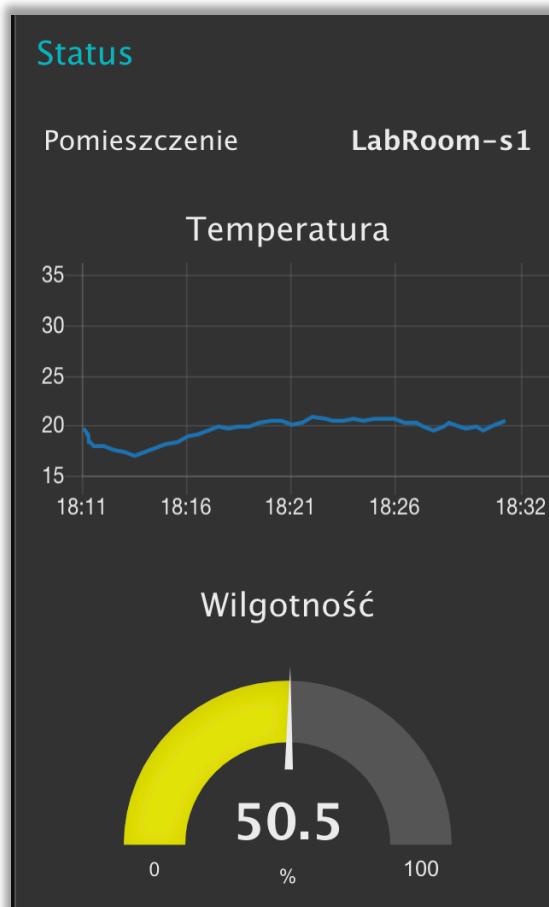
167. Przejdź do zakładki „Site” i ustaw jak poniżej:

168. Przejdź do zakładki „Theme” i ustaw jak poniżej:



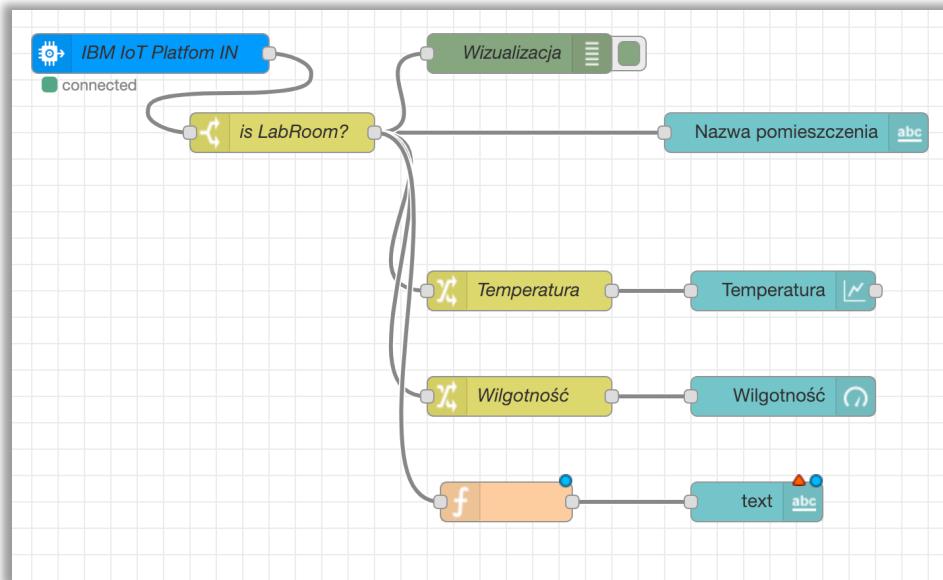
169. Uruchom program przyciskiem „Deploy”.

Przejdź do zakładki w której otworzyłeś widok budowanego interfejsu, odśwież zawartość a następnie sprawdź efekt.



170. Teraz dodamy status otwarcia drzwi, a w późniejszej dodatkowo informacje o numerze karty, którą zostały otwarte drzwi.

Dodaj kolejny węzeł „function” oraz „text”, połącz je jak na rysunku:



171. Kliknij dwa razy na dodanym węźle function i napisz program, który dla poszczególnych wartości pola msg.payload.d.door[A|B|C] wyśle odpowiedni tekst. Pamiętaj, że w oknie „debug” możesz prześledzić treść przesyłanych komunikatów!

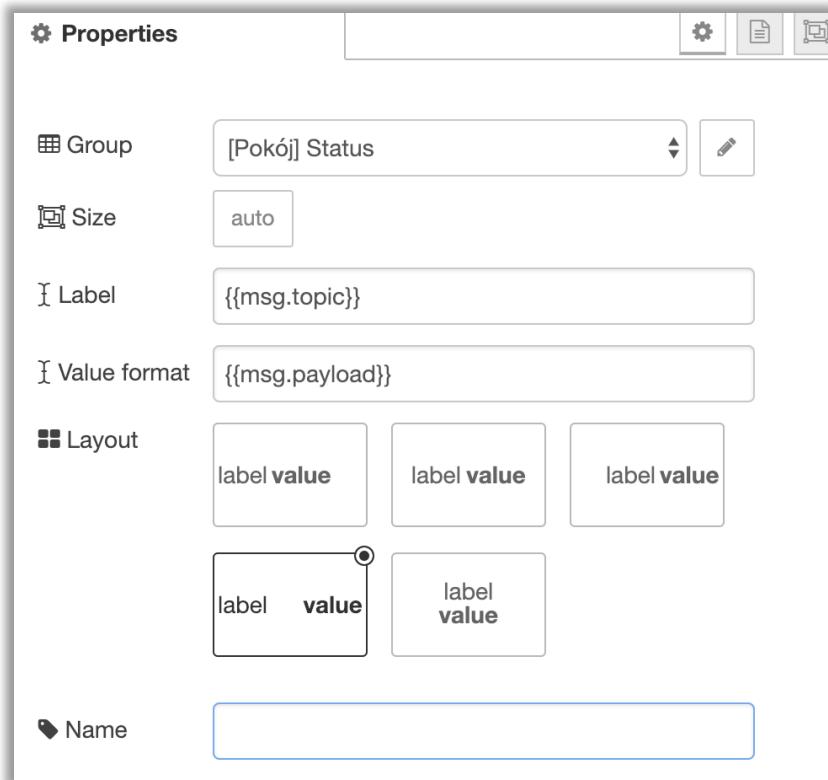
0	Czujnik wyłączony
1	Otwarte
2	Zamknięte

```
var tekst;
var NazwaDrzwi="doorA";

if (! msg.hasOwnProperty('payload')) { return }
if (! msg.payload.hasOwnProperty('d')) { return }
if (! msg.payload.d.hasOwnProperty(NazwaDrzwi)) { return }

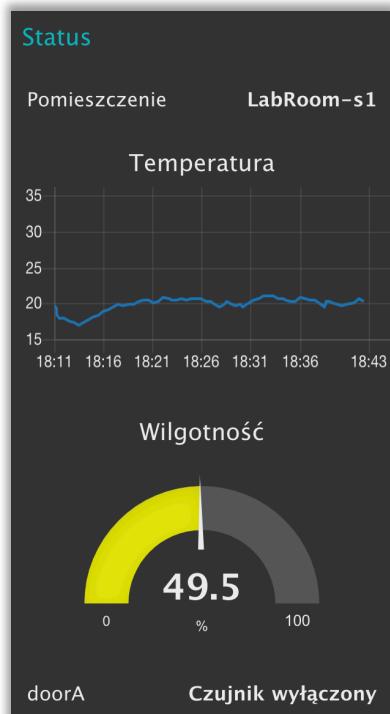
switch (msg.payload.d[NazwaDrzwi]) {
    case 0:
        tekst="Czujnik wyłączony";
        break;
    case 1:
        tekst="Drzwi otwarте";
        break;
    case 2:
        tekst="Drzwi zamknięte";
        break;
    default:
        tekst="Stan nieznany";
        break;
}
return {topic: NazwaDrzwi, payload: tekst};
```

172. Kliknij dwa razy na nowododanym węźle „text” i skonfiguruj go zgodnie z poniższym rysunkiem

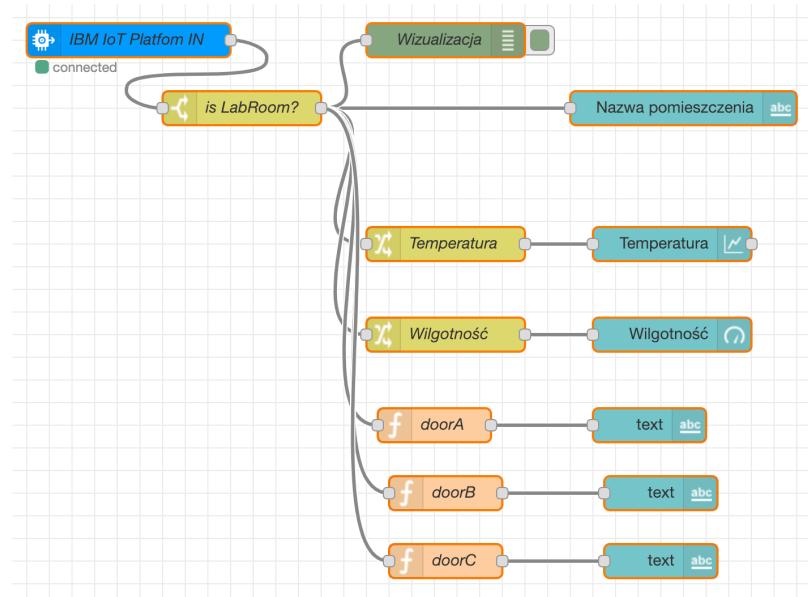


173. Zamknij okno edycji widgetu i uruchom program przyciskiem „Deploy”.

Przejdź do zakładki w której otworzyłeś widok budowanego interfejsu i sprawdź efekt.



174. W podobny sposób dodaj status drzwi doorB i doorC – wykorzystaj tę samą treść programu zamieniając jedynie zawartość zmiennej NazwaDrzwi.



175. Zamknij okno edycji widgetu i uruchom program przyciskiem „Deploy”.
Przejdź do zakładki w której otworzyłeś widok budowanego interfejsu i sprawdź efekt.

15. SYMULOWANIE OTWARCIA I ZAMKNIĘCIA DRZWI

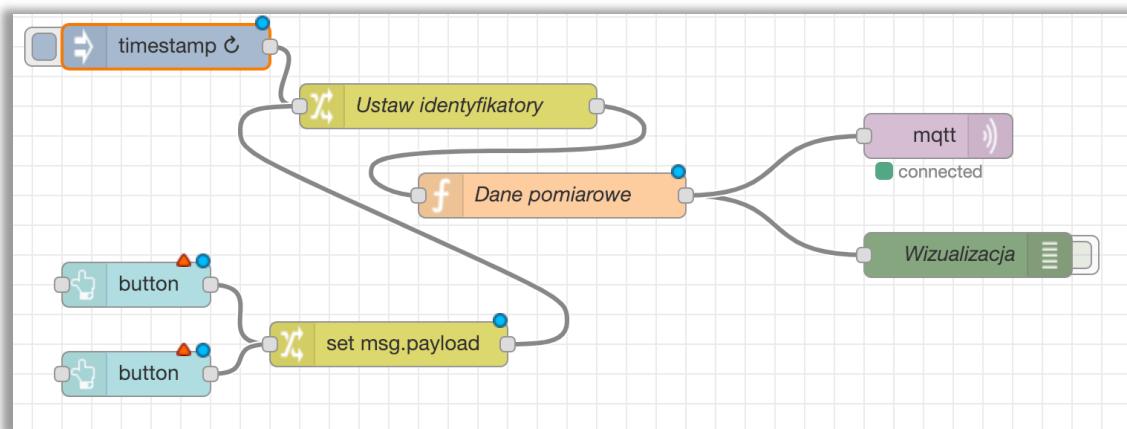
176. W kolejnym kroku rozbudujemy nasz symulator urządzenia IoT (który umieściliśmy w zakładce „Symulator”) o funkcjonalność przesyłania informacji o otwarciu i zamknięciu drzwi.
Przejdź do edycji programu Symulator.



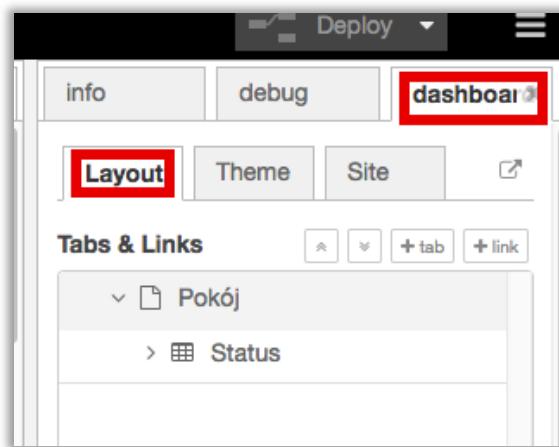
Naszym celem jest dodanie funkcji:

- Aktualizacji zawartości zmiennej przechowującej bieżący stan drzwi (zmienne globalne przepływu o nazwie doorA, doorB, doorC). Zmienne ta będą współdzielone pomiędzy wszystkimi blokami należącymi do programu „Symulator”
- Wysłanie nowego zdarzenia o nazwie „event” do platformy IoT

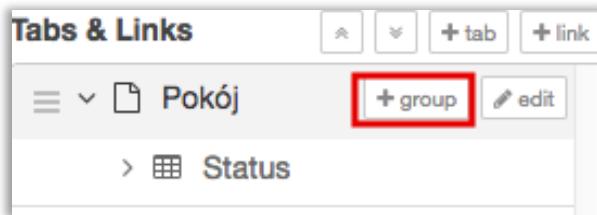
177. Dodaj do ekranu roboczego dwa węzły typu „button” i jeden węzeł „Change”. Będą one symulowały otwieranie i zamykanie drzwi pomieszczenia poprzez ustawianie zmiennych globalnych tego programu, które następnie będą odczytywane podczas wysyłania danych do platformy. Połącz jak poniżej:



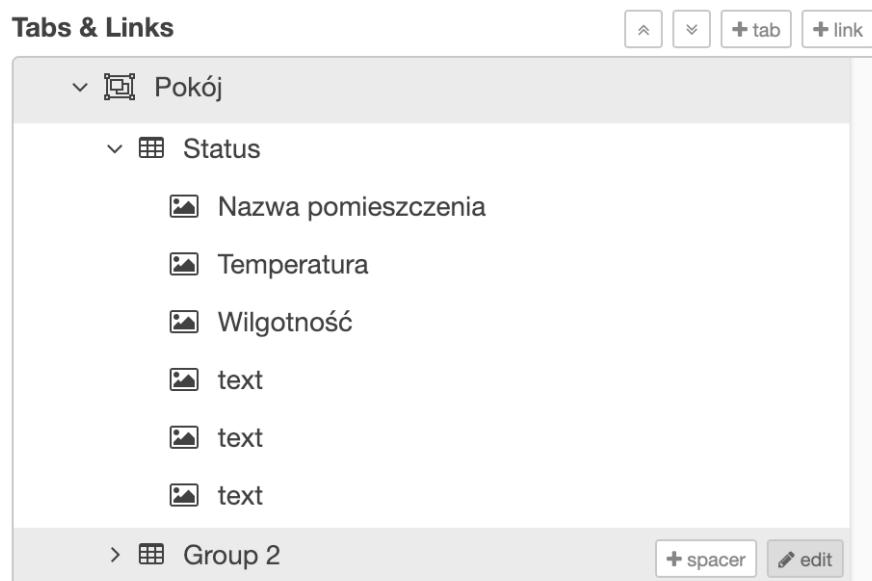
178. Przyciski dodamy do nowej grupy na desktopie Node-Red-Dashboard. Aby ją dodać, znajdź po prawej stronie ekranu zakładkę „dashboard” a pod nią zakładkę „Layout”

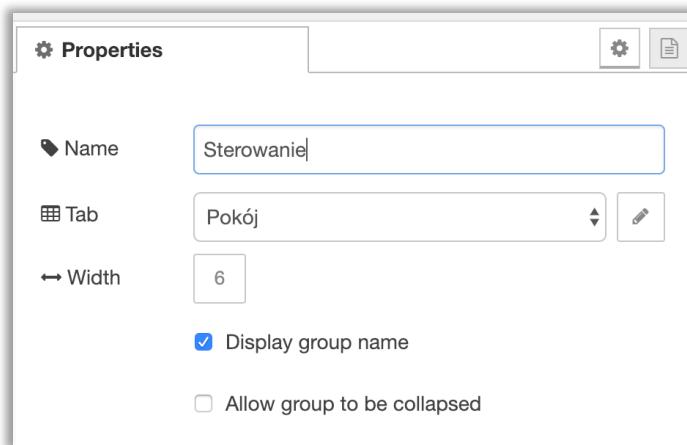


179. Kliknij na nazwie strony (Pokój) i wybierz „+ group”, pojawi się grupa „Group 2”



180. Kliknij na nowoutworzonej grupie, wybierz „edit” i zmień jej nazwę na „Sterowanie”



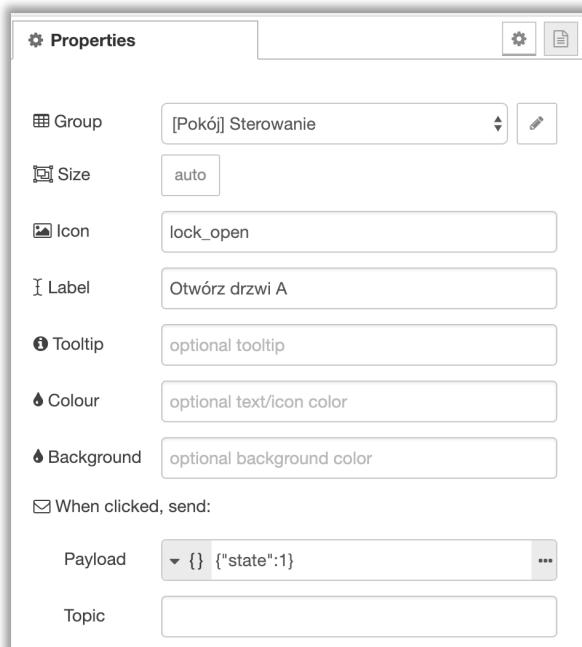


181. Kliknij „Update” aby zaakceptować zmiany. Powróć do edytowania programu symulatora.

182. Kliknij dwukrotnie na pierwszym z dodanych przycisków, a następnie:

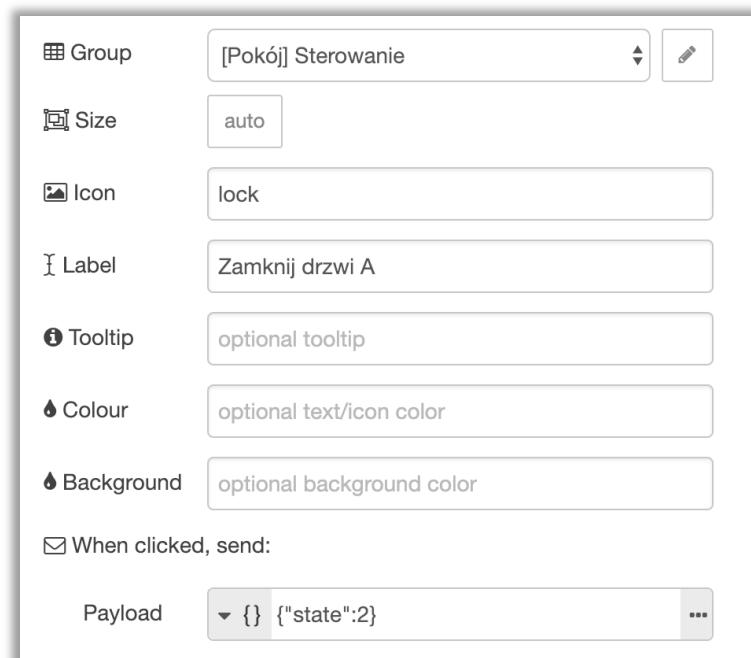
- Dodaj go do grupy „[Pokój] Sterowanie”
- Nadaj nazwę „Otwórz drzwi A”
- Ustaw ikonę „lock_open”
- Ustaw aby po kliknięciu wysyłał obiekt:

| {"state":1}

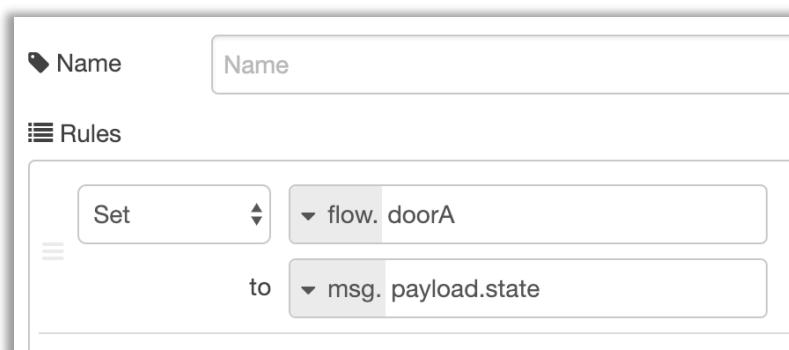


}

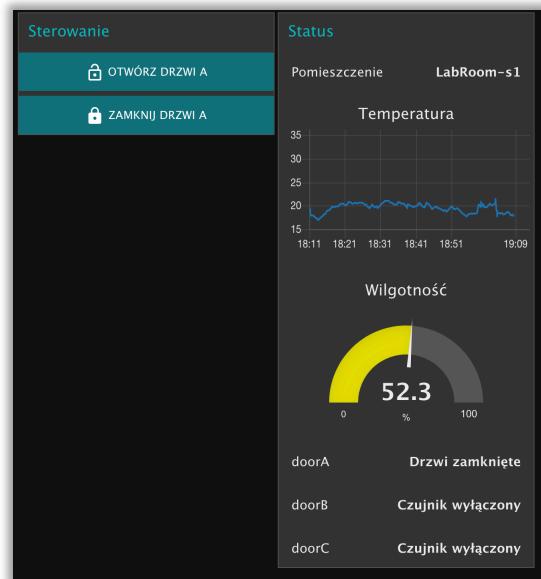
183. Czynności powtóż tworząc przycisk „Zamknij”. Payload dla operacji zamykania drzwi będzie się różnił polem state, które musi być wówczas ustawione na wartość 2 i ikoną – „lock”:



184. Ustaw węzeł „Change” jak poniżej:



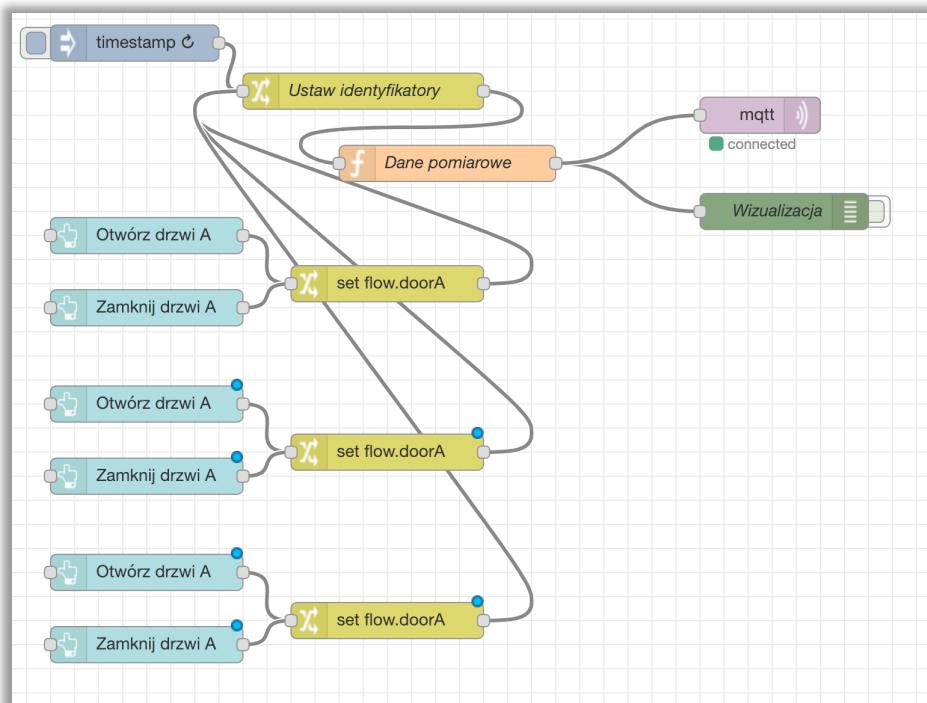
185. Zamknij okno edycji widgetu i uruchom program przyciskiem „Deploy”. Przejdź do zakładki w której otworzyłeś widok budowanego interfejsu i sprawdź efekt.



186. Przejdź do edytora Node-RED

187. Zaznacz myszką dwa dodane węzły button i jeden change, wciśnij Ctrl-C aby je skopiować.

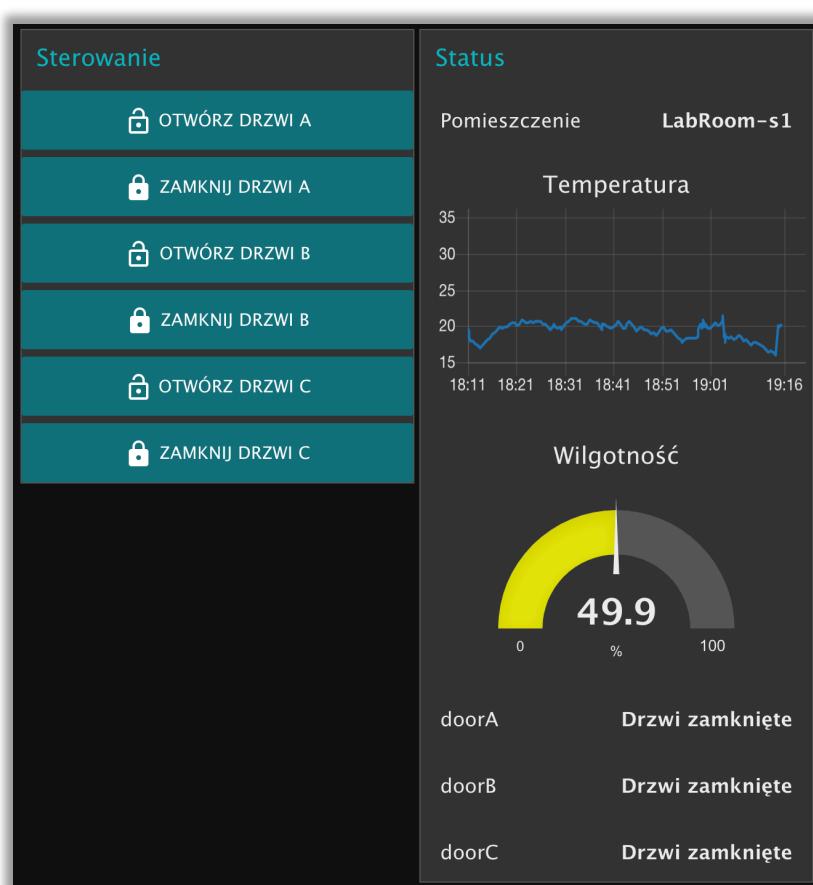
188. Wklej dwukrotnie (Ctrl-V) i połącz jak poniżej



189. Zmodyfikuj wklejone węzły tak, aby wysyłały zdarzenia dla doorB i doorC

190. Zamknij okno edycji widgetu i uruchom program przyciskiem „Deploy”.
Przejdź do zakładki w której otworzyłeś widok budowanego interfejsu i sprawdź efekt.

Sprawdź, czy przyciski zmieniają status odpowiednich drzwi



ZADANIA DODATKOWE – DLA CHĘTNYCH

16. NUMER KARTY

191. Dodaj nowy typ zdarzenia mówiący o tym, jaką kartą zostały otwarte drzwi. Proponowany format komunikatu, wyzwalany przyciskami otwarcia drzwi:

```
"topic":"iot-2/evt/event/fmt/json"  
  
"payload":{  
    "d":{  
        "timestamp":1555321311294,  
        "door":"doorA",  
        "card":"010291120012"  
    }  
}
```

192. Dodaj wizualizację numeru karty którą otwarto drzwi.